College of Computer and Information Sciences

Computer Science Department

"A Multimodal Deep Learning Approach for Fake News Detection Across Text and Visual Modalities"

CSC 497 – Final Report

Prepared by:

Hajar Almeleehan	444200728
Jana Alamer	444200767
Sarah Aljuhani	444200927
Dimah Alotaibi	444201161
Athoog Alsuhaibany	444202989

Supervised by:

Dr. Sarab Almuhaideb

Research project for the degree of Bachelor in Computer Science

Second Semester 1447/1448

I.English Abstract

Fake news, which can be defined as an intentionally made false information to deceive, manipulate, influence people, or damage the reputation of any entity, has increasingly become widespread in recent years. The spread of fake news can cause significant harm to individuals, whether in mental, physical, or health-related aspects. As a result, there is a growing need for automated systems capable of detecting fake news, and deep learning has become the leading trend in this field due to its strong ability to extract meaningful features and improve model performance. Most of the existing research on this topic adopt a unimodal approach, focusing on a single type of data—most commonly text. However, multimodal deep learning approaches, which utilize more than one type of data, are relatively less common. While fake news is often associated with text, it can also occur in the form of manipulated images. Therefore, a multimodal approach, which integrates both textual and visual modalities, is essential for a comprehensive analysis. In this project, we propose a lightweight multimodal architecture for fake news detection that takes the text and the corresponding image as input to effectively identify fake news. In our architecture, MobileNetV3 is used as the image encoder and TinyBERT as the text encoder to extract feature embeddings from each modality. A projection head then maps these two embeddings into a common dimensional space, after which they are fused using appropriate fusion strategies. Finally, the resulting fused vector is passed to a classification layer to generate the final prediction. To evaluate the proposed architecture, experiments will be conducted on the Fakeddit and Twitter MediaEval benchmark datasets, followed by analyzing performance using standard classification metrics such as accuracy, recall, precision, F1-score, the number of parameters, and FLOPs.

II.Arabic Abstract

تُعرَّف الأخبار الزائفة بأنها معلومات مضللة تُنشأ عمدًا بهدف خداع الأفراد أو التلاعب بهم أو التأثير على آرائهم أو الإضرار بسمعة جهةٍ معينة. أصبحت هذه الظاهرة منتشرةً على نطاقٍ واسع في السنوات الأخيرة، مسببةً أضرارًا بالغة على الصعيد النفسي والجسدي والصحي، مما أوجد حاجة متزايدة لتطوير أنظمة مؤتمتة قادرة على اكتشافها بفعالية. ومع التطور المستمر في تقنيات الذكاء الاصطناعي، برز التعلّم العميق كتوجه رئيس في هذا المجال لما يقدّمه من قدرة عالية على استخلاص السمات المميّزة وتحسين أداء النماذج. تركز معظم الدراسات الحالية في هذا المجال على النهج أحادي الصيغة المعتمد على نوع واحد من البيانات، وغالبًا ما تكون البيانات النصية. في المقابل، يهدف النهج متعدّد الصيغ إلى دمج أكثر من نوع من البيانات للوصول إلى تحليل أشمل وأكثر دقة، إلا ان تطبيقاته لا تزال محدودة الاستخدام نسبيًا. ونظرًا إلى أن الأخبار الزائفة لا تقتصر على النصوص فحسب، بل قد تتخذ أيضًا شكل صور مزيفة أو معدّلة، فإن النهج متعدد الصيغ الذي يدمج بين البيانات النصية والبصرية يُعدّ ضروريًا لتحقيق تحليل شامل ودقيق. لذلك، يقدّم هذا البحث نهجًا متعدد الصيغ قائمًا على التعلّم العميق يدمج بين النصوص والصور المرتبطة بها بهدف الكشف الفعّال عن الأخبار الزائفة، بما يُسهم في تعزيز مصداقية المحتوى الرقمي والحد من انتشار المعلومات المضللة. في هذا البحث، نقترح بنية خفيفة الوزن متعددة الصيغ للكشف عن الأخبار الزائفة ، بحيث تستقبل النص والصورة المرتبطة به كمدخلات بهدف الكشف الفعّال عن الأخبار الزائفة. في هذه البنية، تستخدم الشبكة التلافيفية العصبية MobileNetV3 كمرمز للصور، بينما يُستخدم التمثيل المشفر ثنائي الاتجاه الصغير من المحولات TinyBERT كمرمز للنصوص لاستخراج تضمينات الخصائص من كل وسيط. بعد ذلك، تقوم طبقة الاسقاط بمواءمة هذين التضمينين في فضاء بُعدي مشترك. ثم تُدمج التضمينات باستخدام استراتيجيات الدمج المناسبة، وبعدها يُمرَّر المتجه المدمج الناتج إلى طبقة التصنيف لإنتاج التنبؤ النهائي. لتقييم البنية المقترحة، ستُجرى التجارب على مجموعتي البيانات Fakeddit و Twitter MediaEval ، ويتبع ذلك تحليل الأداء باستخدام مقاييس التصنيف الشائعة مثل: الدقة، والاستدعاء، وإحكام ، مقياس إف 1 ، بالإضافة إلى عدد المعاملات وعدد العمليات الحسابية.

Contents

List of Tables

List of Figures

List of Notations

Abbreviation	Definition	Page no.
ANN	Artificial Neural Network	23
BERT	Bidirectional Encoder Representations from Transformers	33
BiLSTM	Bidirectional Long Short-Term Memory Network	53
CLIP	Contrastive Language–Image Pretraining	61
CNN	Convolutional Neural Network	23
DeBERTa	Decoding-enhanced BERT with Disentangled Attention	34
ELECTRA	Efficiently Learning an Encoder that Classifies Token Replacements Accurately	35
GDES	Gradient-Disentangled Embedding Sharing	35
GELU	Gaussian Error Linear Unit	48
GRU	Gated Recurrent Unit	31
LLaMA	Large Language Model Meta AI	61
LoRA	Low-Rank Adaptation	32
LSTM	Long Short-Term Memory Network	31
MLM	Masked Language Model	33
OCR	Optical Character Recognition	54
ReLU	Rectified Linear Unit	48
RNN	Recurrent Neural Network	31
TF-IDF	Term Frequency-Inverse Document Frequency	22
ViT	Vision Transformer	36
VGG	Visual Geometry Group	26

Chapter 1: Introduction

In recent years, the rapid growth of social media and digital communication platforms has transformed the way information circulates, which allows news, whether true or fake, to spread at lightning speed and reach vast audiences. This rapid spread has increased the impact of fake news, which poses significant threats to social trust [1]. According to research [2], fake news spreads six times faster to audiences than true news and has a 70% higher chance of being retweeted than true stories. This significant difference in how quickly fake and true information spreads online highlights the serious real world impact of fake news.

Manual fact-checking is too slow and cannot keep up with the volume of content that is being produced online. Automated approaches, on the other hand, offer the speed and consistency needed to monitor large amounts of data in real time. Such a system will alert users and platforms of potential fake news before it spreads widely.

Furthermore, fake news appears in different formats, including text, images, video, and audio. In fact, in most cases, it integrates more than one modality. As a result, there is a need for an automated fake news detection system capable of analyzing not just a single modality, but multiple modalities in an integrated manner.

Most early fake news detection systems focused on a unimodal approach [3, 4, 5]. Text-based unimodal approaches can capture the contextual characteristics of written content which enable the detection of patterns that may indicate misinformation. Although this approach can identify many deceptive or misleading articles, it overlooks an important element, images, which can reinforce misleading claims and add emotional weight to deceive audiences.

Similarly, image-based unimodal approaches [6, 7] focus exclusively on visual content which allows models to detect manipulated images and other visual cues that may indicate deception. However, this approach is not very effective because it overlooks the text that often accompanies images and it can provide vital clues for evaluating the veracity of the content.

On the other hand, a multimodal approach to fake news detection combines both text and images, allowing for a complete analysis of news content. Unlike unimodal methods that focus on a single modality, multimodal systems understand that fake news often relies

on the combination of text and visuals to seem more convincing. By examining text and images together, multimodal systems can spot inconsistencies that might be overlooked when looking at each modality separately and this can improve detection accuracy.

Several studies [8, 9, 10] have investigated a multimodal approach to the detection of fake news, which combines text and images and uses deep learning models to enhance detection accuracy. These studies consistently show that integrating multiple modalities is more effective than relying on text or images alone, especially for detecting deceptive or misleading content.

Despite progress in multimodal fake news detection, gaps still exist in the current studies. Most existing multimodal fake news detection studies [11, 12] focus on improving accuracy by using large and computationally expensive models. However, these approaches are not efficient, and this underscores the need for lightweight multimodal models. Lightweight models are designed to be fast and efficient by using less parameters and requiring less computation. Examples of lightweight models include MobileNet [13] and GhostNet [14] for images and TinyBERT [15] and MiniLM [16] for text. In addition, many studies did not examine different fusion strategies, which represents an important limitation, since the way text and image features are fused can greatly impact the model's performance. Also, different fusion strategies can capture information from each modality in a unique way, which may lead to better and more accurate fake news detection. Therefore, further research is needed to compare various fusion strategies while maintaining computational efficiency.

1.1 Problem Statement

With information and news rapidly spreading across the media, some of this news is bound to be fabricated. Fake news is a serious problem that can result in consequences in both our personal and academic lives [1]. Traditional machine learning techniques have been previously used to detect fake news in text [17, 18, 19, 20], however, they face significant limitations in handling the challenges of natural language, such as the reliance on sophisticated feature engineering and the inability to fully understand the semantics and contextual meaning of text [21]. Additionally, many instances of fake news are associated with a corresponding image, however, most previous research focused on unimodal approaches that analyze only the textual content of the fake news, without considering the image [8]. Moreover, most existing multimodal fake news detection frameworks that processes both textual and visual information consist of computationally expensive and large models [11, 22]. This emphasizes the need to deploy lightweight multimodal deep learning approaches that take as input both the textual information and the corresponding image of the news item and produce as output a classification label indicating the authenticity of the news. Although the multimodal approach is challenging, as it requires the processing and fusion of text and image data, it is worth pursuing to capture the relationships between modalities and improve the accuracy of fake news detection.

1.2 Goals and Objectives

The main goal of this project is to design, implement, and evaluate a deep learning based multimodal architecture for fake news detection. By leveraging both textual content and the corresponding images, the project aims to overcome the limitations of unimodal approaches and contribute to improving the accuracy of fake news detection systems. In relation to this, we intend to answer our research questions:

- "Can a lightweight architecture be effectively introduced for multimodal fake news detection while achieving competitive performance?"
- "How do different fusion strategies affect the overall performance of multimodal fake news detection approaches?"

To achieve our goal, the objectives of the project are defined as follows:

- Review the related work on unimodal and multimodal fake news detection approaches based on text and image modalities to develop a deeper understanding of existing methods.
- Prepare and clean the Fakeddit [23] and Twitter MediaEval [24] datasets, including text cleaning, image resizing/normalization, handling missing or low-quality images, and ensuring proper data splits to prevent leakage.
- Develop a multimodal architecture by integrating lightweight deep learning models for textual and visual content.
- Test different lightweight deep learning models, such as MobileNet [25], GhostNet [14], TinyBERT [15], and MiniLM [16], to determine the most suitable model for text representation and the most suitable model for image representation.
- Explore different multimodal fusion strategies.
- Evaluate the performance of the multimodal architecture and analyze the results using standard metrics (accuracy, precision, recall, F1-score, FLOPs, number of parameters).
- Compare the proposed multimodal approach with both multimodal and unimodal results reported in prior research studies.
- Highlight the strengths and weaknesses of the proposed framework, and derive solid conclusions on its overall performance.

1.3 Proposed Solution

To address our problem, this research contributes to the detection of fake news by developing a multimodal architecture that leverages both textual and visual modalities. Unlike most previous work that focuses on unimodal textual processing, we propose a multimodal architecture that integrates different deep learning models, dedicated for image and textual processing, to automatically detect fake news in text captions and their corresponding images. This will be done by exploring different deep learning models for image analysis and for text analysis separately and selecting the best model for each modality. The chosen text

and image processing models will then generate embeddings, which will be fused together to form the final multimodal architecture. Furthermore, we will utilize multimodal fake news detection datasets, which includes Fakeddit [23] and Twitter MediaEval [24], that contains both images and their associated text captions. Moreover, to support practical deployment, we aim to implement our architecture using lightweight models, such as GhostNet-V2 [14] and MobileNetV3 [25] for image encoding and TinyBERT [15] and MiniLM [16] for text encoding, and explore different fusion strategies, strengthening our contribution to multimodal research.

1.4 Research Scope

Our research focuses on developing a multimodal system for detecting fake news using text and visual data. The scope of the research is limited to analyzing images along with their associated text captions accompanying news posts and does not include other types of media, such as video or audio. Methods for extracting and combining text and image features will be explored to improve detection accuracy. Moreover, the news will be classified into two categories: fake and real. Accordingly, the research seeks to investigate the effectiveness of a lightweight deep learning multimodal model, combining images and associated text, in achieving competitive performance. In addition, the research explores how different fusion strategies influence the overall performance of multimodal fake news detection models.

1.5 Research Significance

As people increasingly turn to social media as their main source of news, fake news serves as a tool to distort beliefs, emotions, decisions, and trust in media. Developing stronger detection methods is therefore highly important, particularly through multimodal approaches that combine both text and images. A key limitation in current models is the lack of lightweight architectures, as most existing systems are large and computationally expensive, making them difficult to use on mobile or social media platforms. This project addresses that gap by implementing a lightweight multimodal approach that maintains accuracy while reducing model complexity. The significance of this work is that it provides

a more efficient and accessible method of identifying misinformation, offering benefits for future research, practical system development, and broader efforts to protect society from the harmful effects of fake news.

1.6 Ethical and Social Implications

Since misinformation directly affects individuals, communities, and society [1], ethical and social issues related to fake news detection are highly important, particularly in multimodal approaches that integrate both textual and visual content. In this project, privacy is preserved by relying on publicly available datasets, such as Fakeddit [23] and Twitter MediaEval [24], ensuring that no personal or sensitive information is collected. Although fake news detection systems in general could be misused for censorship or narrative control [26], this project is conducted strictly for academic purposes and is designed to be applied responsibly. As deep learning models learn patterns from data, there is a possibility of unintentionally introducing bias that could lead to unfair outcomes. To minimize such risks, the datasets were carefully selected and evaluated. Overall, the project aims to reduce the spread of misinformation while remaining sensitive to these ethical and social concerns.

1.7 Report Organization

This report is divided into several chapters. Chapter 2 presents the background information and fundamental concepts, such as deep learning, that are necessary to better understand our project. Chapter 3 contains the literature review, where a review was conducted on the unimodal and multimodal approaches to fake news detection, and a discussion was done to highlight the main findings of the related work. Chapter 4 provides the proposed architecture that will be adopted in this project. Chapter 5 highlights the overall project pipeline, including the datasets we will utilize, model selection, evaluation, and implementation environment. Lastly, a conclusion chapter is provided to conclude the research done so far.

Chapter 2: Background

This chapter provides an overview of the fundamental background relevant to multimodal deep learning approaches for fake news detection across text and visual modalities. It covers essential data preprocessing techniques for both textual and visual data, highlighting text representation approaches including TF-IDF and embeddings. The chapter also includes an overview of various fusion strategies for integrating information from multiple modalities, and presents the performance metrics typically used to evaluate classification models. Furthermore, the chapter highlights key deep learning methods commonly used in classification tasks.

2.1 Data Preprocessing

Data preprocessing represents a fundamental stage in preparing raw multimodal data for computational analysis. It involves a set of techniques designed to handle noise, inconsistency, and redundancy, thereby improving data quality and enhancing the reliability of analysis as well as the performance of predictive models.

2.1.1 Text Preprocessing

Raw text collected from online platforms often lacks consistency in format and contains noise. Common issues include the presence of emojis, stopwords, and unnecessary tags. To make the row text suitable for analysis, text preprocessing is required. This step ensures that the input is standardized and ready to be effectively used by machine learning models. According to Manning et al. [27], key preprocessing steps include:

- **Data Cleaning:** This step focuses on removing irrelevant elements such as HTML tags, URLs, emojis, and duplication. Cleaning reduces noise and retains meaningful content.
- **Text Normalization:** This step standardizes the format of text by lowercasing, making whitespace consistent, and removing punctuation and diacritics. This helps the model treat similar words uniformly.

- **Text Tokenization:** Tokenization splits the text into smaller units, such as words, subwords, or sentences, which are easier for NLP models to process. For example, the sentence "Natural language processing enables machines to understand human language." can be tokenized as:
  - "Natural", "language", "processing", "enables", "machines", "to", "understand", "human", "language", "."
- **Lemmatization:** Lemmatization reduces words to their base or dictionary form (lemma) while preserving meaning and part of speech. Unlike stemming, which may produce invalid words, lemmatization produces correct words. For example:
  - "running" → "run"
  - "better" → "good"
  - "went" → "go"

2.1.2 Image Preprocessing

Image preprocessing is a crucial stage in preparing raw visual data before feature extraction and model training. It enhances the quality of the input images, reduces noise, and ensures consistency across datasets, thereby improving the performance of machine learning models [28]. This section is structured into two main subsections: basic preprocessing and normalization.

2.1.2.1 Basic preprocessing

Basic preprocessing techniques involve fundamental operations applied to images to ensure uniformity and reduce unwanted variations. These include:

- **Grayscale Conversion:** Grayscale conversion reduces an RGB image into a single intensity channel, simplifying computations while preserving details. This technique is particularly useful when color information is not essential for the task, such as in edge detection or texture analysis. Because the human eye is more sensitive to green,

this channel is given higher weight. By reducing data from three channels to one, computational cost is significantly lowered [28].

- **Resizing and Cropping:** Resizing refers to scaling an image to specific dimensions in order to match the input size required by machine learning architectures, while cropping removes irrelevant regions and focuses on areas of interest. This ensures consistency across the dataset and prevents distortion when dealing with models that expect fixed input sizes [29].

  The resizing operation can be mathematically expressed as shown in (1), where $I'(x', y')$ represents the resized image, $I(x, y)$ is the original image, and s_x and s_y denote the horizontal and vertical scaling factors respectively [28].

$$I'(x', y') = I\left(\frac{x}{S_x}, \frac{y}{S_y}\right) \tag{1}$$

- **Color Jitter:** Color Jittering is an image data augmentation technique that changes the color of the input image randomly. This includes changes in the color tone, light intensity, color intensity, and contrast of the image to avoid lighting bias and improve generalization [30].

- **Noise Reduction:** Noise is an unwanted variation in pixel values introduced during image acquisition or transmission, and reducing it is necessary to preserve the integrity of edges, contours, and other structural details. Two widely used spatial filtering techniques are Gaussian filtering and Median filtering.

  Gaussian filtering smooths the image by averaging neighboring pixel intensities using a gaussian kernel, which effectively reduces gaussian-type noise [28]. In contrast, Median filtering replaces each pixel with the median value of its neighborhood. Therefore, it performs well in removing impulse-type distortions commonly known as salt-and-pepper noise [28].

  These techniques help produce cleaner and more reliable images, making later processes like edge detection, feature extraction, and model training more effective.

2.1.2.2 Normalization

Normalization refers to the process of transforming pixel intensity values into a consistent numerical range or distribution. This step reduces variations across images, ensures numerical stability, and prevents features with larger scales from dominating the training process. By standardizing input data, normalization improves convergence rates and overall performance of deep neural networks [31].

- **Min-Max Normalization:** Min-Max normalization rescales pixel values to a fixed interval, usually $[0, 1]$ or $[-1, 1]$, while preserving their relative relationships [32]. It can be mathematically expressed as shown in (2), where X' represents the normalized value, X is the original pixel intensity, and X_{min} and X_{max} denote the minimum and maximum pixel values within the same image.

$$X' = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{2}$$

- **Z-score Normalization:** Z-score normalization centers data around zero mean and scales it to unit variance, producing values based on statistical distribution, as shown in (3). In this equation, X' is the normalized value, X is the original pixel intensity, μ represents the mean, and σ denotes the standard deviation of the dataset [28].

$$X' = \frac{X - \mu}{\sigma} \tag{3}$$

2.2 Textual Representation

In order for algorithms to understand text, we must first convert it into a numerical representation, commonly referred to as vectors. This transformation, known as text representation, is an essential step that allows algorithms to effectively process and learn patterns from textual data. There are several ways in which we may convert preprocessed text into numerical vectors [33]. Traditional statistical approaches include Bag of words [34], while more advanced approaches involve word embeddings, such as GloVe [35] and Word2Vec [36], and contextual embeddings that rely on textual transformers such as BERT [37].

2.2.1 Traditional Statistical Approaches

Once the text is preprocessed, it needs to be transformed into a numerical form that can be handled by machine learning models. One of the earliest and most widely used approaches is the Bag of Words (BoW) model, introduced by Salton et al. [34]. This approach transforms the text into a vector and each element in the vector corresponds to one word of the vocabulary, where its value shows the number of times the word is repeated in the text. The BoW model ignores grammar and word order, but was one of the first approaches that showed how simple statistical representations could still work well for tasks such as document retrieval and text classification.

In addition to the above approach, Term Frequency–Inverse Document Frequency (TF-IDF) is a widely used technique that assigns weights to terms based on how frequently they appear in a document compared to their occurrence across the entire corpus. The core concept of TF-IDF, first introduced by Spärck Jones [38] and later formalized by Manning et al [27], can be expressed as follows (4):

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \tag{4}$$

where $tf_{i,j}$ denotes the frequency of term i in document j, df_i is the number of documents containing term i, and N is the total number of documents in the collection. This formulation effectively down-weights common terms and highlights informative ones, thereby improving retrieval and classification performance.

2.2.2 Word Embeddings

Moving beyond these traditional approaches, word embeddings are a way to represent words as numerical vectors. Each word is mapped to a unique vector that captures its meaning, context, and similarity to other words. Unlike traditional methods, such as bag of words or TF-IDF, embeddings place similar words closer together in a continuous vector space. They are widely used in natural language processing, with common examples including Word2Vec [36] and Global Vectors (GloVe) [35].

- **Global Vectors (GloVe):** GloVe is a word embedding method that represents words as vectors based on how often they appear together in text. It uses co-occurrence matrix to capture these relationships and then learns word vectors from it. Words that occur in similar contexts are placed together in the vector space. In addition, GloVe produces static embeddings, meaning that each word is assigned a single fixed vector regardless of context.

- **Word2Vec:** Word2Vec is a method that converts words into numerical vectors so that words used in similar contexts have similar vectors. It has two main approaches: CBOW (Continuous Bag-of-Words), which predicts a word based on its surrounding words, and Skip-gram, which predicts the surrounding words from a given word. By learning from large amounts of text, it identifies patterns in how words are used and represents their meanings as vectors, capturing the relationships between words.

Although word embedding approaches such as Word2Vec and GloVe have been effective in representing words as vectors, more advanced approaches, called contextual embeddings, have been developed. These models, such as transformer-based models like BERT, generate embeddings that vary depending on the word's context within a sentence.

2.3 Convolutional Neural Networks

To better understand the concept of convolutional neural networks and deep learning, we must first introduce neural networks, which form their foundation.

Artificial Neural Network (ANN) [39], which was first presented in 1958, is a model that is designed to mimic the human brain. It contains a group of interconnected nodes, called neurons, organized into layers that are arranged in a chain structure, where a node in one layer is connected to every node in the next layer, as shown in Figure 2.1. Each layer has a specific function. The input layer takes the input data and passes it to the hidden layers, where most of the heavy computations are performed. The final layer is the output layer, where it predicts an output depending on whether the task is regression or classification.

Each node in the neural network is initialized with a random set of weights and a bias term, and the goal of neural networks is to learn the best values of these parameters to

make good predictions. At each layer of the network, the inputs are multiplied by their corresponding weights through matrix multiplication, and the bias value is added. Additionally, an activation function, such as tanh or Rectified Linear Unit function (ReLU), is usually applied to the neuron's output to determine which information should be passed through the network and which should be suppressed. To learn, the network first predicts by forward propagation, then uses a loss function to determine how far off the prediction is. Then, backpropagation is done where the error is propagated backward, and an optimization algorithm, such as gradient descent, updates the weight and bias values. This is done several times so the network improves its ability to make predictions [40].

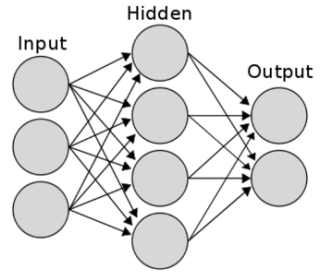


Figure 2.1: The typical layers of a neural network [41].

Convolutional neural networks (CNN) [31] are a specialized form of neural networks that are designed to process data that has a grid-like structure, such as images. A typical CNN architecture consists of three fundamental layer types: the convolutional layer, the pooling layer, and the fully connected layer, as illustrated in Figure 2.2. The convolution layer is the core of the CNN, which is responsible for extracting features such as edges and textures from an input image. This is done by applying filters, called kernels, which are a matrix of trainable weights that slides over the input image to compute a weighted sum of the input pixels, producing feature maps as output. The pooling layer, also known as the down-sampling layer, shrinks the size of the feature maps while maintaining the most important information. Pooling can be performed in several ways, including average pooling, which calculates the average of each region in a feature map, and max pooling, which takes the maximum value of each region in a feature map. Before reaching the fully connected layer, the feature maps that result from the convolutional and pooling layers are flattened into a one-dimensional vector. This vector is then passed to the fully connected layer, which

learns high-level complex patterns to make a decision. The three layers typically use the ReLU activation function to introduce non-linearity so it can learn complex relationships in data. Finally, the output layer (the last fully connected layer) then applies an activation function, such as Sigmoid or Softmax, to generate the final prediction [42, 43].

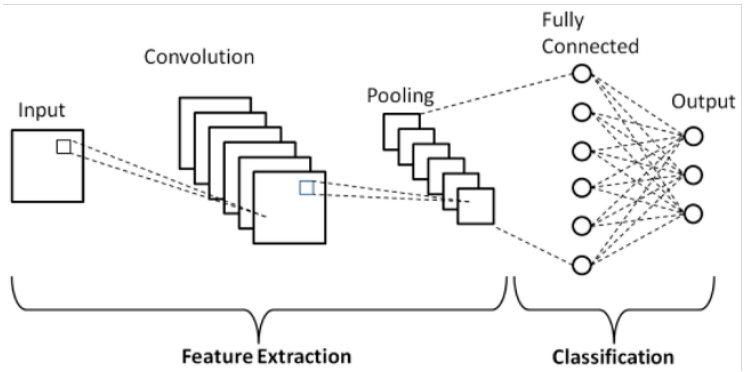


Figure 2.2: Simple structure of a convolutional neural network [44].

Moreover, deep Convolutional Neural Networks are similar to the classical CNNs, however, they have multiple convolutional and pooling layers before the fully connected layer. As a result of the increased depth, deep CNNs can learn more complex features, making them very powerful for the task of image classification [45]. Below, we outline some of the deep CNNs mentioned in the literature.

2.3.1 Residual Network (ResNet)

Residual Network (ResNet) [46] is a deep convolutional neural network that was introduced in 2015 by researchers at Microsoft Research. Before ResNet, researchers tried to improve performance of models by increasing network depth. However, deeper networks had higher training error, which is an effect known as the degradation problem. Therefore, ResNet was introduced primarily as a remedy for the issue of degradation in very deep networks, and it was also helpful in mitigating the issue of vanishing gradient [46]. Vanishing gradient is observed in backpropagation when the gradients become exponentially small while backpropagating through many layers.

The basic building units of the ResNet architecture are the residual blocks, as shown in Figure 2.3. A residual block consists of convolutional layers, batch normalization, a skip connection, and an activation function, usually Rectified Linear Unit (ReLU). Batch normalization [47] normalizes layer inputs per mini-batch, allowing higher learning rates and simpler initialization, while also acting as a regularizer that may eliminate the need for Dropout. In addition, skip connections add the input of a block to the block's transformed output. In other words, the input takes two routes, one goes through the layers which learn transformations of the input, and the other bypasses those layers which carry the input forward unchanged. As a result, gradients can propagate more easily during backpropagation which helps mitigate the vanishing gradient problem. Also, skip connections address degradation by allowing deeper layers to fall back on the input if they do not learn useful transformations. Finally, ReLU activation is applied after the first batch normalization and at the end of the block after the skip connection.

In addition, there are multiple variants of ResNet, such as ResNet-18 and ResNet-34, which differ in the depth of the network.

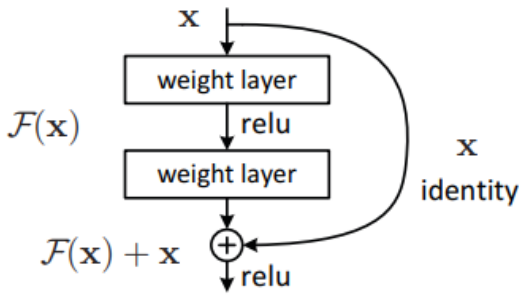


Figure 2.3: Residual block structure [46].

2.3.2 Visual Geometry Group (VGG)

VGG-19 is a convolutional neural network developed by Simonyan and Zisserman in 2014 [48]. Their paper explores how increasing the depth of CNNs affects their performance on large-scale image recognition tasks. Several versions were proposed, including VGG-11, VGG-16, and VGG-19, with VGG-19 being the deepest version consisting of 16 convolutional layers and 3 fully connected layers.

The architecture of VGG-19 is illustrated in Figure 2.4, which shows that the model consists of five convolutional blocks, which use small 3×3 convolutional filters that helped the model reach greater depth. Then, each convolutional block is followed by a Rectified Linear Unit (ReLU) activation function to help the model capture complex features. Next, after each block a 2×2 max pooling layer is applied to reduce the spatial dimensions of the feature maps while keeping the most important features. Finally, after the convolutional blocks extract the features, the feature representation is flattened and then passed through three fully connected layers, where the last fully connected layer acts as an output layer, and a softmax activation function is applied to produce the class probabilities. In addition, the model was trained and evaluated on the ImageNet dataset [49], achieving strong results.

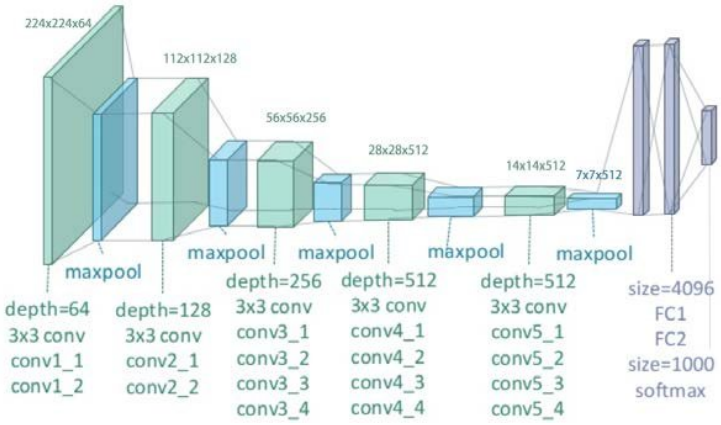


Figure 2.4: Illustration of VGG-19 Architecture [50].

2.3.3 EfficientNet

EfficientNet, which was proposed in 2019 by a team of Google AI researchers [51], is a family of deep CNN image classification models that are trained on the ImageNet dataset. Standard deep CNNs typically scale in only one direction, either deeper by adding layers, wider by adding more filters within the layers, or by increasing the resolution. EfficientNet, however, introduces compound scaling, a concept where the network's depth, which are its layers, width, which are the channels in each layer, and resolution of the input are uniformly scaled together by increasing them simultaneously, as shown in Figure 2.5. Each model in

the EfficientNet family has its own scaling degree, where the more compound scaling is applied, the larger the model becomes and the more computationally expensive it is. This increased scaling is done to achieve better accuracy and improve performance.

The EfficientNet-B0 baseline model has several layers, including an initial convolutional layer that extracts low-level features from the image, followed by several Mobile Inverted Bottleneck Convolutions Blocks (MBConv Blocks) with increasing kernel sizes. These blocks are the core of the EfficientNet model. Each of these blocks takes the feature maps and temporarily increases the number of channels to extract richer and complex features, efficiently processing them. After feature extraction, the number of channels is reduced to make the model more lightweight. Next, a pooling layer is used to shrink each feature map, and lastly, a fully connected layer is used to predict the output. This results in a pretrained model that is highly accurate, however, due to the compound scaling, it is computationally heavy when compared to models such as ResNet.

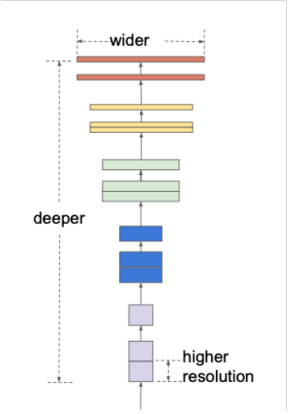


Figure 2.5: Illustration of Compound Scaling [51].

2.3.4 Lightweight Convolutional Neural Networks

Lightweight Convolutional Neural Networks follow the same concept as traditional CNNs. However, they are designed to reduce computational cost, making them ideal for edge devices. The following sections provide an overview of lightweight CNN architectures, namely MobileNet [13] and GhostNetV2 [14].

2.3.4.1 MobileNet

MobileNet [13] was developed by Google in 2017 and was specifically designed for mobile phones and embedded devices. The main idea behind MobileNet is that it splits a convolutional layer into two layers, which are the depthwise convolution and the pointwise convolution, as illustrated in Figure 2.6. First, the 3×3 depthwise convolution layer employs a distinct filter for each input channel that provides spatial features and then a 1×1 pointwise convolutional layer combines outputs across channels. This reduces both computation and number of parameters significantly while maintaining great accuracy.

The early layers capture low-level features such as corners, edges, and textures, whereas the higher layers extract more abstract features, such as object parts or entire objects. As the depth of the network increases, the spatial size of the feature map decreases, but the number of the channel remains high. This structure makes MobileNet able to preserve informative features along the depth of the network.

Moreover, MobileNetV1 has about 4.2 million parameters with 224×224 input and α=1.0. Using the width multiplier reduces this to 2.6M (α=0.75), 1.3M (α=0.5), and 0.5M (α=0.25). MobileNetV2 [52] is more efficient, with around 3.4 million parameters for the standard model. MobileNetV3 [25] adds features such as Squeeze-and-Excitation (SE) [53] modules to perform adaptive channel weighting and uses h-swish [25] activation to enhance efficiency on mobile devices. In addition, it takes advantage from Neural Architecture Search [54]. Although it has slightly more parameters than V2, it remains lightweight. MobileNetV3-Large has about 5.4 million parameters, while MobileNetV3-Small has about 2.9 million parameters.

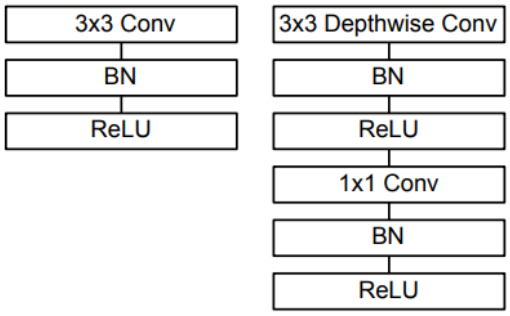


Figure 2.6: Depthwise separable convolution block: decomposition of a standard convolution into depthwise and pointwise layers [13].

2.3.4.2 GhostNetV2

GhostNetV2 [14] is a lightweight CNN model introduced in 2022 used in image classification. Instead of using heavy convolutions to generate every feature map, GhostNetV2 consists of Ghost modules. The Ghost modules, originally introduced in GhostNetV1 [55], consists of a 1×1 pointwise convolution to only generate the essential features, followed by cheap operations, commonly depth-wise convolution, to generate additional features from the essential features. As a result, this allows the Ghost modules to greatly reduce the computational cost, however, it leads to a reduced expressivity in the feature encoding. To address this limitation faced in GhostNetV1, GhostNetV2 was improved by integrating a decoupled fully connected (DFC) attention mechanism, as illustrated in Figure 2.7. In DFC attention, each patch of the feature maps interacts with the patches in the vertical and horizontal directions, producing an attention map. This attention map is then combined with the features generated by the Ghost module through element-wise multiplication, allowing GhostNetV2 to capture long-range dependencies alongside local feature patterns while maintaining computational efficiency.

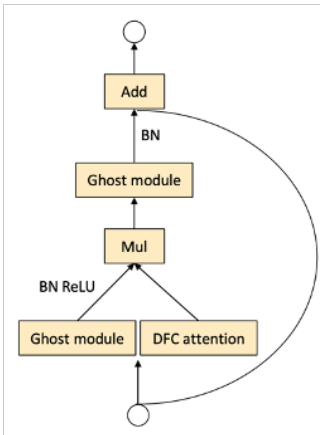


Figure 2.7: Illustration of GhostNetV2 architecture[14].

2.4 Transformers

Before the introduction of Transformers, sequential modeling in NLP primarily relied on Recurrent Neural Networks (RNNs) [56] and their variants such as Long Short-Term Memory (LSTM) [57] and Gated Recurrent Unit (GRU) [58]. RNNs were effective in capturing temporal dependencies, but their sequential nature made training inefficient and limited their ability to model long-range relationships due to vanishing and exploding gradient problems [56]. Although LSTMs and GRUs improved some of these issues, they still struggled with scalability and simultaneous processing. These limitations motivated the development of the Transformer architecture, which replaces recurrence with self-attention mechanisms to efficiently model global dependencies [59].

A Transformer [59] is a neural network model designed to process sequences of data, such as text, using attention instead of older models like recurrent neural networks (RNNs) [56] or long short-term memory networks (LSTMs). Unlike RNNs, a Transformer can process all parts of a sentence simultaneously, making it easier to train and enabling it to understand long-range relationships between words and fully comprehend context. Because of this, it works well for applications such as translating text, summarizing documents, and detecting fake news [59].

A Transformer architecture, as shown in Figure 2.8, consists of two main components: the encoder and the decoder. Each part has layers that use multi-head attention and feed-forward networks. Attention guides the model to pay more attention to the important words in a sentence, while multi-head attention helps the model consider several parts of information simultaneously. Because transformers do not read sentences sequentially like RNNs, positional encodings are used to represent the order of words [59].

In general, the Transformer is a powerful and efficient model that can handle a wide range of natural language processing tasks without relying on traditional sequential networks, using attention mechanisms and being pretrained on large datasets [60].

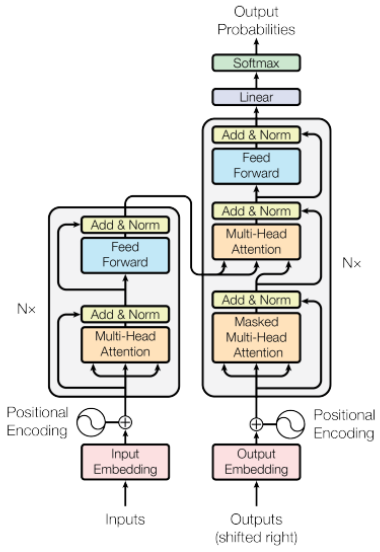


Figure 2.8: The Transformer - Model Architecture [59].

As the field of Transformers evolves, it is considered very computationally expensive to fine-tune Transformers for a specific task. As a solution for this challenge, a number of parameter-efficient methods have been introduced, including Low-Rank Adaptation.

Low-Rank Adaptation (LoRA) [61] is a parameter-efficient fine-tuning technique designed to customize large pretrained Transformers for a specific task. Since Transformers are designed for large applications, it is inefficient to train all transformer parameters for specific tasks. LoRA freezes the original weight matrices ($d \times d$), where d is the dimensionality of

the model, and introduces two new small trainable matrices A and B with a much lower rank r, effectively replacing the weight matrices with $(2d \times r)$ parameters. Compared to full fine-tuning, LoRA can achieve a reduction in the number of trainable parameters by about 10,000 times and reduces the use of GPU memory by about threefold [61]. Unlike other traditional adapters, LoRA does not add additional latency during inference, since the learned low-rank matrices can be merged back into the original weights after training. This makes LoRA an efficient and scalable method for adapting large Transformer models such as BERT and DeBERTa to downstream domains while maintaining competitive performance to full fine-tuning.

2.4.1 Bidirectional Encoder Representations from Transformers (BERT)

BERT belongs to the family of transformer-based language models introduced by Devlin et al. in 2018 [37]. Their study investigated how bidirectional pre-training of transformers can enhance performance on a wide range of natural language processing (NLP) [62] tasks. Two configurations were proposed, namely BERT-base and BERT-large [37], with BERT-large being the deeper version containing 24 transformer encoder layers, 16 attention heads, and a hidden size of 1024, as illustrated in Figure 2.9. The overall architecture of BERT is shown in Figure 2.10 [37], where the network is composed solely of stacked transformer encoder blocks. Each block integrates multi-head self-attention and feed-forward neural networks, along with residual connections and layer normalization to stabilize the training process. Unlike traditional left-to-right or right-to-left models, BERT leverages a masked language modeling (MLM) [37] objective, where a percentage of input tokens are randomly masked and the model is trained to predict the original tokens based on the surrounding context. This helps the model capture bidirectional dependencies in text. Additionally, the model uses next sentence prediction (NSP) [37], a binary classification task that determines whether a given sentence logically follows another. This enables BERT to better understand sentence-level relationships, which is particularly useful for tasks such as question answering and natural language inference.

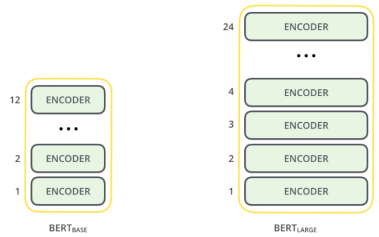


Figure 2.9: Architectural comparison of BERT-base and BERT-large models [63].

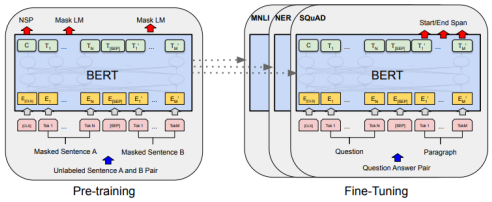


Figure 2.10: Illustration of BERT's pre-training and fine-tuning workflow for various NLP applications [37].

2.4.2 Decoding-enhanced BERT with Disentangled Attention (DeBERTa)

Decoding-enhanced BERT with Disentangled Attention (DeBERTa) was introduced by He et al. in 2021 [64] as an improvement over RoBERTa [65] and the original BERT models. The model improves the attention mechanism by separating the content and position information of words into separate vectors, which helps to better capture dependencies across tokens in the text. In addition, DeBERTa uses an improved mask decoder that enhances the model's accuracy in predicting masked tokens. As illustrated in Figure 2.11, while the original BERT decoding layer processes content and positional embeddings together, DeBERTa adds an extra path that keeps them separate, which improves the understanding of the text and allows the model to perform better on data compared to earlier transformer-based models [64].

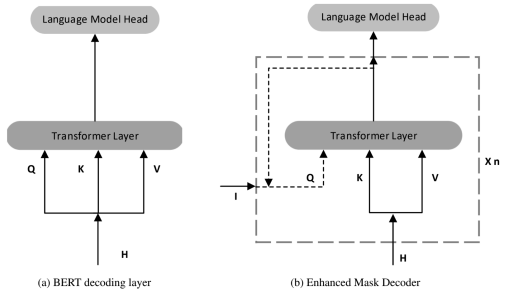


Figure 2.11: The DeBERTa model architecture with enhanced mask decoder [64].

Building on the success of DeBERTa, DeBERTaV3 [66] is an improved version of DeBERTa that uses ELECTRA's Replaced Token Detection (RTD) strategy [67] instead of the original Masked Language Modeling (MLM) objective, making the model learn contextual representations more efficiently. Unlike MLM, where the model hides some words and tries to guess them, RTD randomly replaces some tokens in the text with alternatives that are generated using a generator network. Then, the model must distinguish between the original and replaced tokens. This approach makes the training process stronger and allows the model to train on all input tokens rather than only the masked positions. In addition, DeBERTaV3 introduces gradient-disentangled embedding sharing (GDES) [66], the mechanism is illustrated in Figure 2.12. In this design, the encoder and decoder use the same embeddings, while keeping the gradients separate to prevent interference and redundancy during backpropagation. By separating gradient in this way, GDES reduces redundancy in learned representations and improves training efficiency. Therefore, DeBERTaV3 uses better pretraining strategies and larger datasets, which enable the model to achieve state-of-the-art results on a range of NLP tasks such as question answering and text classification.

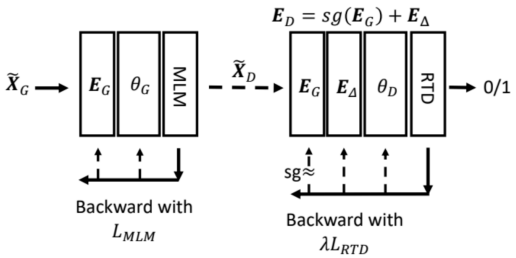


Figure 2.12: The DeBERTaV3 model architecture with GDES [66].

2.4.3 Vision Transformers (ViT)

Vision Transformers (ViTs) [68] use the Transformer architecture from natural language processing (NLP) for image recognition tasks. ViTs are highly effective at modeling global relationships across the entire image, which allows them to capture long-range dependencies that CNNs may miss. In addition, ViTs achieve strong performance on large datasets, however, when they are trained on smaller datasets, they often underperform when compared to CNNs. This is mainly because ViTs do not capture local image patterns, which is highly important for effective learning from small datasets.

In a ViT, the input image is first divided into non-overlapping patches, which are then flattened into one-dimensional vectors known as tokens. Each token is linearly projected into a lower-dimensional vector to reduce dimensionality while keeping the necessary features. Furthermore, positional embeddings are added to the tokens to indicate the spatial location of each patch within the image which ensures that spatial relationships are maintained. After that, all token embeddings are fed into the Transformer encoder. The first layer in the Transformer encoder is self-attention, which allows each patch to gather information from all other patches and hence captures dependencies between them. Following self-attention, a feed forward network is applied independently to each token which models complex and nonlinear relationships among the patches. Finally, an MLP head will produce the final output based on the global representation learned by the encoder. The model overview of the Vision Transformer is shown in Figure 2.13.

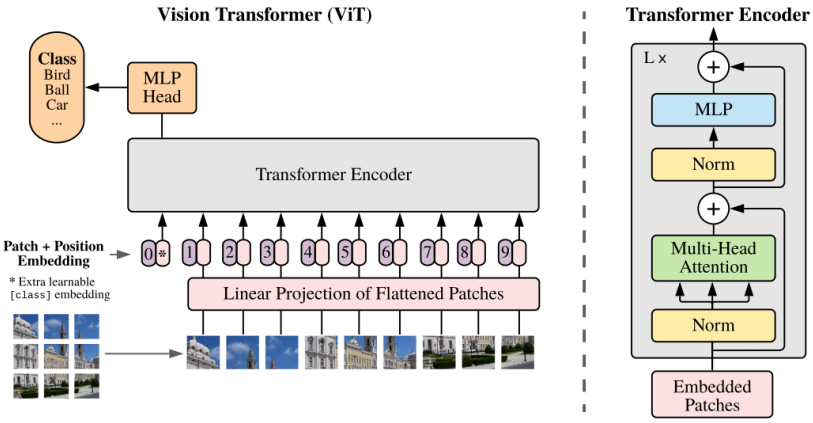


Figure 2.13: Vision Transformer Model Overview [68].

2.4.4 Lightweight Transformers

Traditional transformer models have a high computational cost, making them difficult to deploy on edge devices. Lightweight transformers, which are more efficient, were developed to address this issue. Below, three lightweight transformers are presented, where two are text-based and one is imaged-based.

2.4.4.1 TinyBERT

TinyBERT is a lightweight model that was introduced in 2020 [15] to reduce computational expenses through a proposed transformer distillation method that enables smaller student models to learn from larger teacher models. In large language models (LLMs) distillation is the process of transferring knowledge from a teacher model to a student model [15].

Both BERT and TinyBERT share the same transformer architecture but TinyBERT has only 4 layers, making it easy to apply the transformer distillation, which is performed in three steps. Firstly, the main addition was the transformer layer distillation, which includes attention-based distillation that helps the student learn the semantic knowledge from the teacher and hidden-states-based distillation, which aims to minimize the mean squared error of the hidden representations of the teacher and the student. In addition, embedding layer distillation and prediction layer distillation are used to help the student mimic the teacher's performance. Furthermore, a learning method was introduced, which includes the general distillation and the task-specific distillation, as illustrated in Figure 2.14. The first stage is the general distillation stage, where the student learns from the teacher BERT without fine-tuning using a large-scale text corpus. For the task-specific distillation, data augmentation is performed on a task-specific dataset, then transformer distillation is performed using a fine-tuned BERT. After this stage, we obtain a smaller version of BERT, called TinyBERT, which has the ability to perform specific tasks and generalize well. With only 14.5 M, TinyBERT manages to achieve 96.8% of BERT's performance while being faster and lighter.

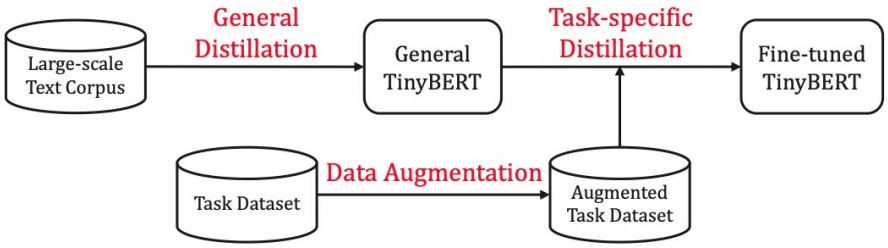


Figure 2.14: The illustration of TinyBERT learning. [15].

2.4.4.2 MiniLM

MiniLM is a compact and effective Transformer-based model introduced in 2020 [16] that compresses large pre-trained language models via a novel deep self-attention distillation framework. Unlike prior approaches that rely on layer-to-layer mapping, MiniLM trains the student by deeply mimicking the self-attention module of the teacher's last Transformer layer , as illustrated in Figure 2.15. The model learns not only the attention scores between queries and keys but also the relationships between values, known as value-relations, that capture more detailed information. This value-relation transfer produces relation matrices of fixed size, which lets student models adopt arbitrary hidden dimensions without introducing extra projection parameters. The method also benefits from using a teacher-assistant when the size gap is large, bridging performance between the original teacher and smaller students. Experiments proved that the 6-layer MiniLM, which was distilled from BERT-BASE, performs almost like the teacher model on common benchmarks such as GLUE [69] and SQuAD 2.0 [70], but with fewer parameters and faster inference, about twice as fast. Overall, deep self-attention distillation offers a simple, flexible, and computationally efficient path for task-agnostic compression of pre-trained Transformers.

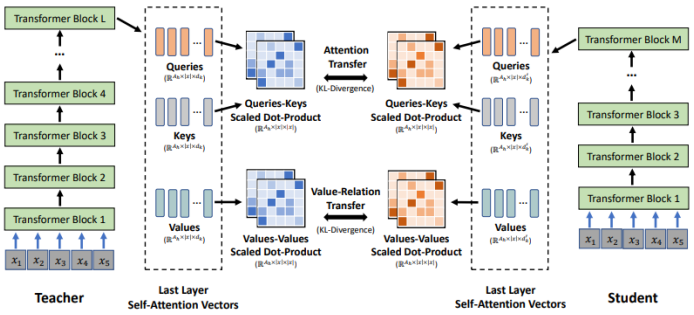


Figure 2.15: Overview of the MiniLM deep self-attention distillation framework .[16].

2.4.4.3 MobileViT-v2

MobileViT-v2 [71] was introduced in 2023 by Mehta and Rastegari as an improved and more efficient version of MobileViT. The original MobileViT [72] is a hybrid model that combines the benefits of CNNs and Vision Transformers, where the CNNs part focuses on local features, while the global context is handled by transformers. However, since it still uses normal self-attention, it compares every patch with all the other patches, which makes it slow and not suitable for mobile devices.

MobileViT-v2 solves this limitation by introducing separable self-attention. This attention uses simple element-wise operations, which is a light and fast type of attention, while still keeping global awareness of patches, as shown in Figure 2.16. Separable self-attention reduces the complexity from quadratic $O(k^2)$ to linear $O(k)$. This makes the model capable of understanding the whole image with much less computation, while remaining efficient. In experiments, it showed higher accuracy and was about three times faster than the first version, making it more practical for vision tasks on mobile and edge devices.

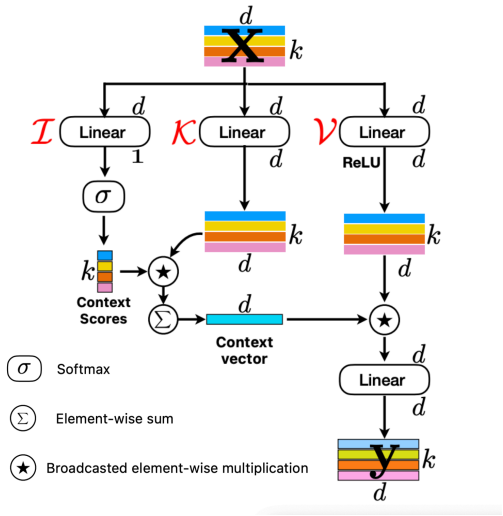


Figure 2.16: MobileViT-v2 architecture with separable self-attention [71].

2.5 Siamese Neural Network

Siamese Neural Network [73] was first introduced in the 1990s by Bromley et al. for the task of human signature verification. It is generally used to compare between two inputs by measuring their similarities. The general concept of the network is that it consists of two branches, each having identical architectures and shared weights. Each of the two branches receive an input that is mapped into a shared feature space, where their representations are compared by using a metric, such as cosine similarity, to measure the similarity between the two inputs. In contrast, in a Pseudo-Siamese Network [74, 75], which is a more general form of Siamese Neural Network, the two branches do not share the same weights. Each of the branch may consist of different neural networks, such as CNN on one branch and an RNN on the other, allowing them to be more flexible and applicable to a wider range of applications.

2.6 Attention Mechanisms

Attention mechanisms enable models to identify and emphasize the most relevant spatial regions and channels within feature maps, while reducing the impact of less useful information. These modules enhance feature representations efficiently without adding significant computational complexity, making them particularly effective for tasks such as image classification. There are different kinds of attention mechanisms proposed in literature [76], in this section, we report Squeeze-and-Excitation [53] as it is used in our project. The Squeeze-and-Excitation (SE) mechanism belongs to the channel attention category, the Convolutional Block Attention Module (CBAM) [77] combines channel and spatial attention, and Coordinate Attention (CA) [78] improves spatial attention by integrating positional information.

The Squeeze-and-Excitation (SE) attention mechanism was introduced by Hu et al. [53] to improve the performance of CNNs by modeling channel-wise dependencies. The SE block adaptively adjusts feature responses through two main operations, squeeze and excitation. In the squeeze step, global average pooling is applied to each feature map to obtain a channel descriptor that captures global spatial information. In the excitation step the descriptor is used to generate weights for each channel. Furthermore, the mechanism consists of two fully connected (FC) layers. The first FC layer reduces the channel dimensions to limit the model's complexity, and then the second FC layer restores the dimensions to the original channel count before generating the final weights. In the end, the weights are then multiplied with the original feature maps, producing higher weights to the more important channels. In addition, the SE block can be integrated into existing architectures, such as ResNet and Inception [79], improving their overall performance.

2.7 Fusion Strategies

In order to create a multimodal framework, fusion is required to fuse the image features and the textual features in the architecture. Several fusion techniques exist that differ in how the different modalities are combined, they include traditional fusion approaches such as feature-level fusion and decision-level fusion, and a more sophisticated attention-based fusion approach [80, 81] .

2.7.1 Feature-level fusion

Feature-level fusion is considered to be an early fusion approach that merges the features extracted from the different modalities before the classification layer into a single representation, as shown in Figure 2.17. The single representation is the fused vector that contains the combined features from the different modalities. Several feature-level fusion methods exist to combine the features, they include concatenation and element-wise addition. Concatenation fusion simply combines the text and image embeddings by stacking the text and image embeddings into one extended feature vector that preserves the full content of both modalities [82]. This straightforward method allows the classifier to learn how the two modalities relate to each other using the combined features [82].

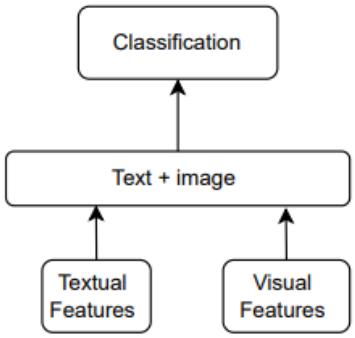


Figure 2.17: Illustration of feature-level fusion [80].

Element-wise maximum fusion compares the two feature vectors and selects the larger value at each dimension, which helps highlight the most informative signals from either modality [83]. This technique strengthens strong features and reduces the impact of weak or noisy ones, making the multimodal prediction more reliable [83].

Additionally, among feature-level fusion methods, element-wise multiplication fusion (Hadamard product), has gained attention in the multimodal field for its ability to combine information from different modalities efficiently. In this strategy, instead of simply joining features together, corresponding values in the text and image feature vectors are multiplied. This highlights the parts where both modalities are strong, allowing the model to better capture semantic consistency between them [84].

To make the Hadamard product more adaptive, recent works such as [84] include element-wise summation as an additional feature-level fusion method. In this approach, corresponding elements from the two modality embeddings are added to form a single vector. This operation allows the model to integrate complementary information from both modalities, rather than relying on strong shared features highlighted by the Hadamard product. In contrast [85] uses element-wise sum to maintain the original feature dimensionality, avoiding the dimensional expansion typically caused by concatenation.

Another feature-level fusion method used to compare the two embeddings is the absolute difference [73]. This method takes the difference between the text and image feature vectors and then applies the absolute value. This allows the fusion step to highlight the mismatch between the two modalities, which can be helpful to identify the contradiction of the image and the text.

Table 2.1: Summary of feature-level fusion strategies

Fusion Strategy	**Feature**
Concatenation	Combines the two embeddings by merging them into a single extended feature vector. This preserves the full information.
Element-wise Maximum	Compares the two embeddings and selects the larger value at each dimension. This highlights the most informative signals and reduces noisy features.
Hadamard Product	Multiplies the corresponding values of the two embeddings at each dimension. Used to capture semantic consistency.
Element-wise Summation	Adds the corresponding values of the two embeddings. This maintain the original feature dimensionality.
Absolute Difference	Computes the absolute value of the element-wise difference between embeddings. Used to capture disagreements between embeddings.

2.7.2 Decision-level fusion

Decision level fusion, also referred to as late fusion, combines the predictions of the models classification layer as decisions or confidence scores, rather than merging their features at an earlier stage, as shown in Figure 2.18. This approach is widely used in multimodal systems because of its simplicity and efficiency [86]. Two common strategies are:

- Maximum fusion: picks the class that has the highest confidence score from all the classifiers. This approach works well when one of the classifiers is usually very confident and makes good predictions [87].

- Average fusion: the final decision score for each class is obtained by averaging the confidence scores across all classifiers. This method balances the influence of individual models and improves stability [88].

Decision level fusion is simple and does not require much computation. It is often used in multimodal applications such as image analysis and scene understanding, because it can improve the overall prediction accuracy by combining the strengths of different classifiers [89].

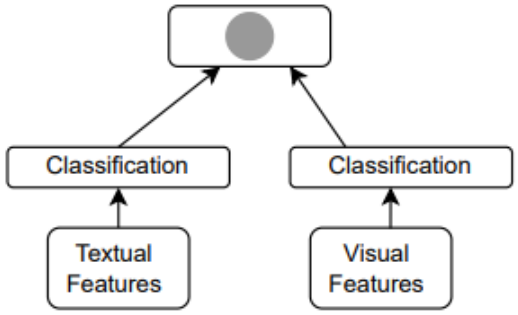


Figure 2.18: Illustration of decision-level fusion [80].

2.7.3 Attention-based fusion

Attention-based fusion is an approach that models the interaction between modalities, such as text and images, by assigning weights to features, allowing the model to focus on the most informative parts of the data. This provides a way to capture relationships between modalities. Different attention based fusion methods include:

- Cross-modal attention: Cross-modal attention fusion is an attention-based fusion strategy that merges information from several (mainly two) modalities by allowing one modality to focus on the other's important information while ignoring the unnecessary part. In cross-modal attention, the Query (Q) tensor comes from one modality,

which specifies what that modality is focusing on. The Key (K) tensor, which contains the information to be matched with, and the Value (V) tensor, which contains the actual content that will be retrieved when a match is found, comes from the other modality. These values are used to compute the attention between the two modalities, as shown in equation (5).

$$\text{Attention} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

After computing the attention, the modality from which the Query tensor comes from gets an updated vector representation that combines its original features with the most important part from the other modality, where the Key and Value tensor are obtained [59, 81]. Once the vectors of each modality are aware of each other, these vectors go through several layers of the neural network before finally being fused using a feature-level fusion strategy [90].

- Co-attention: Co-attention is a fusion mechanism that models the relationship between modalities by updating their representations with information from each other. In this process, each modality is refined using the other as context, and the two are fused in a bidirectional manner [91]. Different versions of co-attention have been proposed. In parallel co-attention, the two modalities process each other's features at the same time. Each modality generates weights that determine the importance of the other's features, allowing both representations to be updated simultaneously [91]. In alternating co-attention, the interaction happens in sequence rather than simultaneously. One modality is refined using the other, and in the next step the updated representation is used to guide the first modality, allowing the two to progressively refine each other [91]. Dense co-attention first involves computing a comprehensive similarity matrix that explicitly models the relevance between every feature element in one modality and every feature element in the other. This ensures a symmetric interaction across the entire feature space [92]. Finally, stacked co-attention involves stacking multiple co-attention layers on top of each other. The features produced by one co-attention layer are passed as input to the next layer, allowing the model to perform deep reasoning on multiple stages, and progressively learn more complex

joint representations [93]. Despite its usefulness, co-attention has been less widely used in recent multimodal approaches, where cross-modal attention has become more dominant [94, 95].

2.8 Datasets

Various datasets have been introduced for the study of multimodal fake news detection. Fakeddit [23], Weibo [96], and Twitter MediaEval [24] are among the most widely used datasets. A summary of these datasets is provided in Table 2.2, which includes only multimodal instances composed of text and images. For our research, we focus on Fakeddit, which provides a diverse and representative benchmark for fake news detection. In addition, we will also utilize the Twitter MediaEval dataset [24].

Fakeddit dataset is a multimodal fake news dataset that contains more than 1 million instances. It includes both text and image data that are collected from Reddit, a popular social media platform. The dataset supports multiple levels of classification with 2-way, 3-way, and 6-way labeling schemes. For the 2-way classification, instances are labeled with either true or fake. For the 3-way classification, the instances are labeled with completely true, fake with false text, or fake and contains text that is true. Finally, for the 6-way classification, the instances are labeled with true, satire/parody, misleading content, manipulated content, false connection, or imposter content. Moreover, the dataset contains 682,661 multimodal samples, 268,908 true samples, and 413,753 fake samples.

Weibo [96] is a Chinese multimodal fake news dataset collected from Sina Weibo's[1] [97] official community for rumor debunking. It includes 9,528 events, with 4,749 labeled as rumors and 4,779 as non-rumors and it was collected from 2012 to 2016. In Weibo's rumor debunking system, users can report suspicious posts. These posts are then reviewed by a committee of users to determine whether they are real or false. Posts that are confirmed as false are labeled as rumors, while posts that are confirmed as real by the Xinhua News Agency [98], a trusted news source in China, are labeled as non-rumors.

The Twitter MediaEval dataset was introduced for the MediaEval 2016 Verifying Multimedia Use task [24]. It contains tweets related to 17 major world events and includes

[1] http://www.weibo.com/

text, images or videos, and metadata. The posts were collected using keywords related to the 17 events and a visual near-duplicate search strategy to retrieve real and fake images shared during these events. Each post is labeled as real or fake depending on whether the associated image or video relates accurately to the event described in the text.

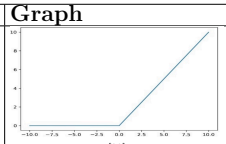
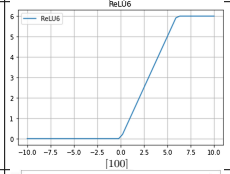
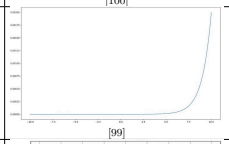
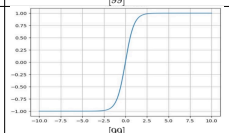
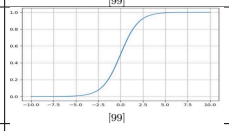
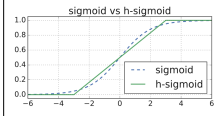
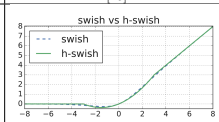
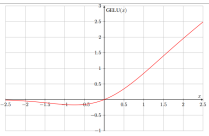
Table 2.2: Summary of commonly used datasets for fake news detection, reporting only text-image multimodal instances.

Dataset	**No. of Multi-modal Instances**	**No. of Classes**	**Class Distribution**	**Date**
Fakeddit	682,661	2, 3, 6	Fake samples= 413,753 (60.6 %) and real samples= 268,908 (39.4 %).	2020
Weibo	9,528	2	Rumors= 4,749 (49.8%) and non-rumors= 4,779 (50.2%).	2017
Twitter MediaEval	15,821	2	Real= 6,225 (38.3%) and fake= 9,596 (60.7%)	2016

2.9 Activation Functions

Activation functions play an important role in neural networks by introducing non-linearity, which allows the model to capture complex patterns in data. They also shape how information flows through the layers and influence training stability and predictive performance. Table 2.3 summarizes the most widely used activation functions, highlighting their equations and graphs.

Table 2.3: Activation Functions along with their mathematical equations and graphical representations.

Activation Function	Equation	Graph	Remarks
Rectified Linear Unit (ReLU)	$f(x) = \max(0, x)$	 [99]	Outputs the maximum between zero and the input value.
ReLU6	$ReLU6(x) = \min(\max(0, x), 6)$	 [100]	Both hard-sigmoid and hard-swish rely on the clipped linear behavior of ReLU6.
Softmax	$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$	 [99]	Converts input into probabilities that sum to 1. Often used for multi-class classification.
Tanh	$f(x) = \tanh(x)$	 [99]	Maps input values to the range $[-1, 1]$.
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	 [99]	Maps input values to the range $[0, 1]$. Often used for binary classification.
Hard-Sigmoid (h-Sigmoid)	$h\text{-}Sigmoid(x) = \frac{\text{ReLU6}(x+3)}{6}$	 [25]	Cheap approximation of sigmoid. This is the formula used in MobileNetV3.
Hard-Swish (h-swish)	$h\text{-}swish(x) = x \cdot \frac{\text{ReLU6}(x+3)}{6}$	 [25]	Efficient approximation of the swish activation. It is designed for mobile models, providing similar performance benefits with lower computational cost.
Gaussian Error Linear Unit (GELU)	$f(x) = 0.5x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)\right)\right)$	 [101]	Maps input values (negative, zero, and positive) into a smooth non-linearity that preserves small negatives and gradually amplifies positives.

2.10 Performance Metrics

To evaluate the performance of the models, several performance measures are used to ensure that different aspects of performance are examined. These measures usually rely on the classifier's ability to make correct predictions. The confusion matrix of a binary classifier, as shown in Table 2.4, indicates the number of true and false predictions made by the classifier. This representation aids in understanding the different proposed performance measures [102].

Table 2.4: Confusion Matrix of a binary classifier

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

- True Positive (TP): The classifier correctly predicts a positive outcome.
- True Negative (TN): The classifier correctly predicts a negative outcome.
- False Positive (FP): The classifier predicts a positive outcome, however, the prediction is incorrect.
- False Negative (FN): The classifier predicts a negative outcome, however, the prediction is incorrect.

Below are some of the common performance measures that are often computed using the confusion matrix.

Accuracy (Acc) is used to measure how often the classifier makes correct predictions, as shown in (6). High accuracy indicates the classifier makes many correct predictions while low accuracy means it makes few correct predictions [102].

$$Acc = \frac{TP + TN}{TP + TN + FN + FP} \tag{6}$$

Recall (R), also known as sensitivity or True Positive Rate, is used to measure how often the classifier correctly predicts the positive outcome when the actual outcome is positive, as shown in (7) [102].

$$R = \frac{TP}{TP + FN} \tag{7}$$

Precision (P) is used to measure how often the classifier correctly predicts the positive outcomes among all the outcomes predicted as positive, as shown in (8) [102].

$$P = \frac{TP}{TP + FP} \tag{8}$$

F1- Score ($F1$) is the weighted average of precision and recall, as shown in (9). It is mainly useful when the dataset is imbalanced [102].

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{9}$$

Receiver Operating Characteristic (ROC) curve is a graph that is used to envision the performance of the classifier by plotting the True Positive Rate against the False Positive Rate across different threshold values [102].

Area Under the Curve (AUC) refers to the area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate against the false positive rate at different thresholds. The AUC value ranges from 0 to 1, where a score of 0.5 indicates performance equivalent to random guessing and a score of 1 indicates perfect classification [103].

Floating Point Operations (FLOPs) measure the computational cost of a model by counting the total number of arithmetic operations required for a single forward pass. The total FLOPs of a network are obtained by summing the operations of all its layers [104].

Parameters refer to the total number of learnable weights and biases in a neural network. They are the numerical values that the model updates during training to learn patterns from the data. The total number of parameters is obtained by summing the parameters of all layers in the neural network [105].

2.11 Summary

In this chapter, various foundational concepts supporting fake news detection were introduced. The preprocessing techniques for text and images were first briefly discussed to prepare data. Then, the different text feature representation methods were explained. The chapter also reviewed key deep learning architectures such as CNNs and Transformers, highlighting notable models like BERT and DeBERTa, as well as lightweight Transformers like MobileViT-v2. Furthermore, the chapter explained attention mechanisms in general, with a focus on Squeeze-and-Excitation. In addition, Siamese Neural Networks were discussed, and the chapter introduced popular multimodal fake news datasets such as Fakeddit. The chapter also examined different fusion strategies used to combine text and image information, and these strategies were categorized into three main types: feature-level, decision-level, and attention-based fusion.

Chapter 3: Literature Review

In this chapter, we begin by reviewing unimodal approaches to fake news detection, which mainly focus on textual features, with limited work addressing visual features. We then review multimodal approaches that integrate the textual information with the corresponding image of the news item, demonstrating how they improve detection accuracy. Additionally, a review was conducted on several lightweight frameworks and on different applications of Siamese Network.

3.1 Unimodal Approaches

Several studies have investigated unimodal approaches to fake news detection, where either text or image modalities are used to determine if the news item is real or fake, below we review a selection of these works.

3.1.1 Textual-based Unimodal Approaches

Multiple studies have been conducted to detect fake news using a textual-based unimodal approach, where only text is taken as input and classified as either fake or real news, as summarized in Table 3.1.

Stance detection, which is defined as the automatic identification of the relationship between two pieces of text, represents a key approach in addressing the challenge of fake news detection when implemented using deep learning methods. In this study [106], the Fake News Challenge (FNC-1) dataset [107] is used, which categorizes headline–article pairs into four stance classes: Agree, Disagree, Discuss, and Unrelated. The model was trained using three different variations of neural networks. The Term Frequency-Inverse Document Frequency (TF-IDF) vector representation combined with a dense neural network achieved an accuracy of 94.31%, which represents the highest performance and outperformed existing models by about 2.5%. The Bag-of-Words (BoW) representation with a dense neural network obtained an accuracy of 89.23%. Surprisingly, when pretrained embeddings such as Word2Vec were used as input to a dense neural network, the model's accuracy dropped to 75.67%.

Similarly, Subhash et al. [108] compared several deep learning models with different embeddings, including Word2Vec, GloVe, and FastText for fake news detection using the Kaggle Fake News dataset (US 2016 election articles) [109]. Their Bidirectional Encoder Representations from Transformers (BERT)–based [37] model achieved state-of-the-art accuracy of 99.2%, outperforming Recurrent Neural Network (RNN) [56], Long Short-Term Memory (LSTM) [57], Gated Recurrent Unit (GRU) [58], and Convolutional Neural Network (CNN) [31] variants, but remained limited to text-only input.

Additionally, Sharma and Sahu [4] proposed a deep learning framework for fake news detection using the ISOT Fake News dataset [110]. After preprocessing the data through converting to lowercase, cleaning, removing stop words, stemming, the features were extracted using Bag of Words, TF-IDF, tokenization, padding, and GloVe embeddings. The framework was evaluated with CNN, LSTM, Bidirectional Long Short-Term Memory (BiLSTM) [111], and a hybrid CNN-BiLSTM model. The results showed that CNN and LSTM achieved the best accuracy at 83%, with LSTM giving the highest precision and F1-score, while BiLSTM and CNN-BiLSTM performed slightly lower at 82% and 81% respectively. The hybrid model was reported to not improve performance, suggesting that while deep learning methods are effective, further optimization is required to achieve the higher accuracy levels reported in other studies.

Bhattacharjee et al. [3] explored the use of a Long Short-Term Memory (LSTM) model to detect fake news, specifically focusing on news titles rather than the entire article. This research employed the ISOT dataset, a publicly available dataset composed of two files, True.csv and Fake.csv, which were combined and preprocessed for analysis. The preprocessed data was transformed into a numerical format through word embedding, where one-hot encoding is applied. The authors then trained the LSTM model on the dataset, which involved using k-fold cross-validation and the Adam optimizer to improve accuracy. The results show that the LSTM model achieved an overall accuracy of 93.5%, outperforming traditional machine learning methods such as logistic regression [112] and Support Vector Machine (SVM) [113], which achieved an accuracy of 85.3% and 86.7% respectively. Although this demonstrates the effectiveness of the LSTM model, challenges remain in handling ambiguous content.

Meanwhile, Kamble et al. [114] conducted a comparative study using the WELFake

dataset [115], evaluating both machine learning and deep learning approaches. Among the machine learning models tested, Extreme Gradient Boosting (XGBoost) [116] achieved the highest accuracy of 96.71%. For deep learning methods with GloVe embeddings, the BiLSTM model achieved the best performance with 91.58% accuracy. However, transformer-based models such as BERT [37] outperformed all others, reaching 98.95% accuracy. Compared to traditional and earlier deep learning models, transformers achieved the best reported performance in this study.

Furthermore, Salem et al. [5] proposed a hybrid CNN-LSTM model for fake news detection which aims to improve the accuracy of detecting image-based fake news by leveraging the strengths of both models. Their approach analyzed only the text embedded within the images rather than the image itself. In addition, the Fake News Classification Image Dataset [117], which contains both textual and visual information, was utilized. For data preprocessing, Optical Character Recognition (OCR) was used for the detection of textual content embedded in images, which allows visual text to be transformed into machine-readable text. Following that, the extracted text was cleaned to standardize it by converting all letters to lowercase, removing non-alphabetic characters, and replacing multiple consecutive whitespaces with a single space. Next, the text was tokenized and padded so that all sequences had the same length. In the CNN and LSTM models, there was an embedding layer with an input size of 512 as a representation of the vocabulary size and an output size of 16 as a representation of the size of the embedding vector for each token. For the hybrid CNN+ LSTM model, the embedding layer had an input of 512 and an output of 128. In addition, label encoding was performed to convert the categorical labels into numerical representations suitable for binary classification, with 'Real' mapped to 1 and 'Fake' mapped to 0. After preprocessing, the CNN was used to detect local patterns and key phrases within the text and the LSTM captured broader contextual and sequential relationships. The hybrid model combined the two advantages by first passing the text through CNN layers to determine the important features and then passing them as inputs to the LSTM for contextual representation. As a result, the hybrid model achieved an accuracy of 90.98%, which outperforms both the standalone CNN model with an accuracy of 86.67% and the standalone LSTM model with an accuracy of 85.49%.

Table 3.1: Summary of Textual-based Unimodal Related Work

Article	Ref.	Year	Feature Representation/ Preprocessing	Model/ Algorithm	Dataset	Metric used	Results	Key Insight
Fake News Detection: A Deep Learning Approach	[106]	2018	Bag of Words, TF-IDF, and Word2Vec	Bow + DNN, TF-IDF + DNN, Pretrained Word2Vec embeddings + DNN, Proposed Hybrid Model: TF-IDF vectors + engineered cosine similarity features fed into DNN.	Fake News Challenge (FNC-1)	Acc	TF-IDF + Dense NN accuracy = 94.31%, BoW + Dense NN accuracy = 89.23%, Word2Vec + Dense NN accuracy = 75.67%	Pretrained embeddings (Word2Vec) underperformed compared to simpler TF-IDF, using cosine similarity lead to better performance, data imbalance affect minority stance categories (Agree, Disagree)
Fake News Detection Using Deep Learning and Transformer-Based Model	[108]	2023	Glove, Word2Vec, FastText, BERT	RNN, LSTM, BiLSTM, GRU, Bi-GRU, CNN-LSTM, CNN-BiLSTM, FastText, BERT	Kaggle dataset on the 2016 U.S. Presidential Elections	Acc, P, R, F1	BERT accuracy= 99.20% F1-score = 0.99	BERT-based approach achieved state-of-the-art performance,Glove embeddings outperformed Word2Vec, RNN was the weakest
Fake News Detection using Deep Learning Based Approach	[4]	2023	Lowercase, Clean, Remove stopwords, Stemming, Tokenization+Padding, BoW, TF-IDF, GloVe	CNN, LSTM, BiLSTM, hybrid CNN-BiLSTM	ISOT Fake News dataset	Acc, P, R, F1	CNN accuracy = 83.0%, LSTM accuracy = 83.0%, BiLSTM accuracy = 82.0%, CNN–BiLSTM accuracy = 81.0%	Text-only deep models ≈83% acc, hybrid underperforms, tuning/richer features needed
Title-Based Fake News Detection Using LSTM	[3]	2024	One-hot encoding	LSTM, Logistic Regression, SVM	ISOT Fake News dataset	Acc, P, R, F1	LSTM accuracy= 93.5%, LR accuracy of 85.3%, SVM accuracy = 87.1%	LSTM outperforms traditional machine learning models on title-based fake news detection
Fake News Detection Using Machine Learning and Deep Learning	[114]	2024	Bag of words, TF-IDF, Word2Vec, GloVe, BERT	LR, KNN, SVM, DT, RF, XGB, SGD, MNB, GBM, LSTM, BiLSTM, BERT	WELFake	Acc, P, R, F1	BERT achieved 98.95% accuracy	Deep learning models outperformed machine learning models, BERT achieved the highest accuracy
Detecting Fake News Images Using a Hybrid CNN-LSTM Architecture	[5]	2025	OCR is used to extract text embedded in images	CNN, LSTM, Hybrid CNN-LSTM	Fake News Classification Image Dataset	Acc, P, R, F1, ROC AUC, PR AUC, and MCC	LSTM only accuracy = 85.49%, CNN only accuracy = 86.67%, Hybrid CNN-LSTM accuracy = 90.98%	The hybrid CNN-LSTM model outperforms standalone CNN and LSTM models in detecting fake news images

3.1.2 Image-based Unimodal Approaches

Although most unimodal approaches focused on the detection of fake news in textual information only, some have explored unimodal approaches that strictly analyze images to detect fake news, as shown in Table 3.2.

Some of the images that accompany fake news are manipulated. These manipulations can be done in different ways, such as copying and moving parts, removing objects, or combining multiple images. These changes can make images misleading and less reliable. Recent studies focus on detecting such manipulations by looking at both visible features and hidden patterns in the images.

Peng Zhou et al. [7] proposed a two-stream Faster R-CNN [118] network for detecting manipulated regions in images. The RGB stream captures visual tampering artifacts such as unnatural edges and contrast differences, while the noise stream extracts local noise inconsistencies using a steganalysis rich model (SRM) [119] filter layer. Features from both streams are fused via bilinear pooling to improve detection accuracy. The model was trained on a synthetic COCO-based dataset [120] and fine-tuned on four standard datasets (NIST16 [121], CASIA [122], COVER [123], Columbia[124]). Results show that the combined RGB-N network outperforms individual streams and baseline methods such as ELA, NOI1, CFA1, MFCN, and J-LSTM, which represent earlier approaches for image forensics, ranging from error-level and noise analysis to deep learning-based models, achieving robustness to resizing and JPEG compression, and effectively distinguishing splicing, removal, and copy-move manipulations. The study emphasizes the complementary role of RGB and noise features in capturing rich tampering evidence, though further enhancements could improve the detection of copy-move manipulations.

S. Alqurashi et al. [6] used a CNN model to robustly process the images and detect the status of the image, whether fake or real within the context of news. The authors first preprocessed the images using the CASIA v2.0 dataset, where three different preprocessing techniques are applied to compare between them and select the best one. The three techniques used are: Error Level Analysis, which computes the difference in compression error levels between the original and the re-saved image, Noise analysis, which detects changes in the natural noise patterns that often occurs when manipulating the image, and lastly, Gra-

dient analysis, which examines the brightness across the image to reveal any inconsistencies. The CNN model was then constructed using several layers, including two convolutional layers, one max-pooling layer, one dropout layer, one flatten layer, and two dense layers. After training the model on the dataset, the results show that the dataset in which the images were preprocessed using Error Level Analysis achieved an accuracy of 97.6%, outperforming the Noise analysis and Gradient analysis preprocessing techniques, which achieved an accuracy of 85.8% and 93.7% respectively. Although the model shows high accuracy, the study emphasized that detecting fake news solely from image requires a large and more diverse dataset in order to be effective.

Table 3.2: Summary of Image-based Unimodal Related Work

Article	**Ref.**	**Year**	**Feature Representation/ Preprocessing**	**Model/ Algorithm**	**Dataset**	**Metric used**	**Results**	**Key Insight**
Learning Rich Features for Image Manipulation Detection	[7]	2018	RGB stream (contrast, boundaries) + Noise stream (SRM filter features)	Two-stream Faster R-CNN with bilinear pooling	Synthetic COCO dataset (pre-training), NIST16, CASIA v2.0, COVER, Columbia	Pixel-level F1-score, AUC	Outperformed traditional methods (ELA, NOI1, CFA1). AUC: 0.937 (NIST16), 0.858 (Columbia), 0.817 (COVER), 0.795 (CASIA). Higher F1-score than baselines	Fusion of RGB and noise features captures complementary tampering artifacts and achieves state-of-the-art performance.
Fake Image Detection in Fake news using Convolutional Neural Network	[6]	2025	Three preprocessing methods are done on separate versions of the dataset: ELA, Noise analysis, Gradient analysis	CNN	CASIA V2.0	Acc, P, R, F1	accuracy using ELA: 97.6% accuracy using Noise analysis: 85.8% accuracy using Gradient analysis: 93.7%	Detecing fake news based on manipulated images alone requires a very large dataset. ELA preprocessing outperformed other preprocessing techniques.

3.2 Multimodal Approaches

Since much of the fake news circulating the media utilizes both text and images, researchers have shifted towards a multimodal approach, where both text and image are leveraged to improve classification accuracy.

3.2.1 Traditional Fusion Approaches

In the following studies, the proposed multimodal architectures were constructed using traditional fusion techniques, such as feature-level and decision-level fusion approaches, aimed to merge information from the textual and visual modalities into a unified representation. Common methods include concatenation fusion, where feature vectors from each modality are combined into a single vector, and maximum fusion, which selects the class with the highest confidence score from all the classifiers. Key details of these studies are provided in Table 3.3.

Singhal et al. [125] designed SpotFake, a multimodal fake news detection framework. In this study, BERT [37] was used to extract features from text, while a pretrained Visual Geometry Group 19-layer network (VGG-19) [48] was used to extract features from images. The embeddings that result from these encoders go through a fully connected layer to map to a common dimension of 32. Next, these embeddings are fused using the feature-level concatenation approach. The unified representation then through a fully connected classification layer to predict the final output. Moreover, this study utilized two common multimodal fake news datasets, Twitter MediaEval [24] and the Chinese dataset Weibo [96]. The experiments shows that SpotFake outperforms several existing models, achieving an accuracy of 77.7% on the twitter dataset and 89.2% on the Weibo dataset. The paper concluded that although SpotFake achieves a strong performance, further advancements could be done by working with longer length articles and experimenting with different fusion strategies.

The study in [8] compared the performance of multimodal fake news detection with multiple unimodal models, utilizing the Fakeddit dataset [23]. After basic preprocessing, texts are tokenized, lemmatized, and mapped to integer sequences, which are then padded to equal length. An embedding layer then converts each token into a word vector, with the embedding matrix initialized using both random initialization and the pretrained GloVe

embeddings. For the multimodal approach, two CNNs are employed, one for text processing and the other for image processing, whose output vectors are concatenated to create a joint representation. As a result, the multimodal model reached an accuracy of 87%, outperforming the unimodal CNN, LSTM+CNN, and BERT models evaluated in the same study. These results highlight the advantages of combining text and visual information for improved fake news detection.

Moreover, Hamed et al. [126] presented a hybrid fusion framework for Text and Image Modalities (HF-TIM) that fuses BERT-based text features with VGG-19 visual features using a combination of early and late fusion strategies. Subsequently, the fused features are further integrated using a meta-learning classifier, which produces the final results. The model achieved 93.4% accuracy on the Fakeddit dataset, outperforming a range of baseline techniques and marking an important step forward in the field. Hybrid fusion was shown to maintain modality-specific attributes while simultaneously modeling cross-modal dependencies, ultimately improving classification reliability and accuracy. However, this approach struggles with fake content, highlighting the need for improved fusion strategies and larger, more diverse multimodal datasets.

Similarly, Uppada et al. [9] proposed a multimodal model to detect fake news on social media by combining textual and visual information. The authors tested different combinations of text and image encoders before constructing their architecture. Based on the experimental results, the chosen framework extracts text features using BERT and image features using the Xception architecture [127], while also including image popularity and polarity to catch the emotions and the context. Two fusion strategies, maximum fusion and concatenate fusion, were employed to integrate text and image features for final classification. Experiments on the Fakeddit dataset demonstrated that the multimodal approach outperformed text-only and image-only models, achieving an accuracy of 91.94% and an F1-score of approximately 93%, highlighting the benefit of leveraging multiple modalities for fake news detection.

Additionally, Singh et al. [128] introduced a stacked ensemble-based multimodal framework (SEMI-FND) for fake news detection using the Twitter MediaEval [24] and Weibo Corpus [96] datasets, integrating different models for both text and images. For the visual features, Neural Architecture Search Network (NasNet) mobile [129] was used, while a

stacked ensemble of BERT and Efficiently Learning an Encoder that Classifies Token Replacements Accurately ELECTRA [67] was used for text. The framework achieved 85.8% accuracy on the Twitter dataset and 86.83% on the Weibo dataset. Compared with recent studies, these results are high while also reducing the framework's complexity. The study concluded that stacked ensembling improves accuracy and speed for multimodal fake news detection.

Moreover, Wang et al. [12] developed a Cross-Image Semantic Fusion (CISF) based multimodal model to classify news as real or fake, where it combines text with several images rather than just a single image. This framework uses the BERT model to process the textual content of the news, and the VGG-19 model to capture the image features. To employ the Cross-Image Semantic Fusion, the authors designed an algorithm that applies the attention mechanism when several images are passed as input, to allow images to exchange information. The process is repeated over several iterations, and as a result, the images are blended into one global representation that captures the overall meaning of the images. Finally, the architecture contains a classification layer that concatenates the text vector with the global image vector and applies the SoftMax activation function to predict the class of the news. The model was then trained on two benchmark datasets, Weibo A [96] and Weibo B [130], each containing a collection of fake and real news with text and images. As a result, when compared to unimodal approaches and other multimodal approaches in the existing literature, the CISF based model demonstrates a strong performance. However, it relies on multiple images associated with the fake news articles, which is not the case most of the time, and it requires high computational power.

Moreover, Lin et al. [85] proposed a multimodal fake news detection framework that uses BERT-Base as the textual encoder, while image features are extracted using ResNet-50 [46]. The model was evaluated on two datasets, GossipCop [131] and Fakeddit. The authors compared three fusion strategies: early fusion using concatenation and element-wise addition, joint fusion using element-wise multiplication with summation pooling, and late fusion using averaging with an ensemble learning method employing CatBoost [132] and XGBoost [116]. The results showed that multimodal fusion outperformed unimodal baselines, with the joint and late fusion setup achieving the best performance, reaching 85% accuracy and 83% F1-score on GossipCop and 90% accuracy with F1-scores up to 90% on

Fakeddit.

Similarly, Mohawesh et al. [22] utilized two benchmark datasets, Twitter MediaEval [24] and Weibo [96], for the detection of multimodal fake news. The text underwent minimal preprocessing, however, the textual information in the Weibo NER dataset was translated from Chinese into English using Google Translate. The images were resized and normalized using min-max normalization. The authors leveraged two different models for processing the textual information of the news item: Sentence-BERT (SBERT) [133] and Decoding-enhanced BERT with Disentangled Attention (DeBERTa) [64]. Additionally, Residual Network (ResNet-50) was used to extract the visual features of the associated image. After the models were fine-tuned on the dataset, feature-level fusion was performed using concatenation fusion, where the SBERT and DeBERT embeddings were fused. Then the unified text embedding was concatenated with the image embeddings extracted by the ResNet-50 model. The resulting multimodal framework was then passed as input to the classification layer, and the final model was trained using the Adam optimizer and categorical cross-entropy loss function. The results show that the proposed multimodal architecture outperformed existing models in literature that used the same datasets, where it achieved an accuracy of 87.4% on the Twitter dataset and an accuracy of 88.3% on the Weibo NER dataset. Although the model shows promising results, it has a high computational cost when compared to other multimodal approaches.

Moreover, Mura et al. [11] adapted Themis, which is a modular neural network originally designed for meme classification, to the task of fake news detection. For this study, the multimodal datasets, Fakeddit and ReCOVery [134], were utilized. To address data scarcity and class imbalance, two data augmentation techniques were applied, which are Text Synonyms and Image Transformations (TSIT) and MixGen. TSIT increase the diversity of text by replacing words with synonyms and applying lemmatization, while also adding variety to images through random transformations. In addition, MixGen generates samples by merging pairs of text and images. Both augmentation methods were applied exclusively to the minority class the ReCOVery dataset to reduce imbalance while preserving semantic meaning. Themis integrates a Contrastive Language–Image Pretraining (CLIP) [135] , Vision Transformer (ViT) [68] image encoder, and a TinyLLama (Large Language Model Meta AI) [136] text encoder whose outputs are fused by a Token Merger Module to preserve the most

relevant multimodal features. Furthermore, to enhance efficiency, Low-Rank Adaptation (LoRA) [61] is used to fine-tune attention mechanisms, while weight freezing maintains the knowledge of pre-trained models by keeping all layers of CLIP, ViT, and TinyLLaMA fixed. Finally, the combined multimodal representations are used to produce a binary prediction which classify content as either fake or real. The experimental evaluation tested various customizations of the Themis architecture on the Fakeddit and ReCOVery datasets. The configurations evaluated on the Fakeddit dataset included the Standard Themis baseline, LoRA, Merge-tokens + LoRA, CLIP ViT Large + LoRA, TSIT + LoRA, and MixGen + LoRA. For the ReCOVery dataset, which suffers from class imbalance, experiments focused on augmentation for the minority class and the configurations included the Standard Themis baseline, LoRA, TSIT + LoRA applied to the minority class, MixGen + LoRA applied to the minority class, CLIP ViT Large + MixGen + LoRA, and MixGen + LoRA with additional class balancing. In the experiments, Merge-tokens + LoRA achieved the best results on the Fakeddit dataset, with an accuracy of 80.2%. On the ReCOVery dataset, TSIT + LoRA achieved the highest performance, with an accuracy of 97.5% which outperforms existing models.

Table 3.3: Summary of Traditional Fusion Multimodal apporaches Related Work

Article	Ref.	Year	Model/ Algorithm	Dataset	Metric used	Results	Fusion Strategy	Key Insight
SpotFake: A Multi-modal Framework for Fake News Detection	[125]	2019	Text Encoder: BERT, Image Encoder: VGG-19	Twitter MediaEval and Weibo	Acc, P, R, F1	Achieved an accuracy of 77.7% on Twitter MediaEval dataset and 89.2% on the weibo dataset	Feature-level fusion: Concatenation fusion	SpotFake outperforms several existing models and further advancments could be done by exploring fusion strategies.
Multimodal Fake News Detection	[8]	2022	Unimodal approach: CNN, BiLSTM + CNN, and BERT-Base Multimodal approach: Separate CNNs for text and image.	Fakeddit	Acc, P, R, F1	The multimodal approach outperforms unimodal approaches, achieving an accuracy of 87% compared to 78% with text-only BERT	Feature- level fusion: Concatenation fusion	Integrating textual and visual information enhances the accuracy of detection, with multimodal approaches demonstrating better performance compared to unimodal ones.
Improving Data Fusion for Fake News Detection: A Hybrid Fusion Approach for Unimodal and Multimodal Data	[126]	2022	MFND-HF-TIM (stacking ensemble with BERT, VGG-19, Softmax + meta-learning classifier)	Fakeddit	Acc, P, R, F1	Achieved 93.4% accuracy, F1-score up to 0.965 (true content). Outperformed baseline multimodal fusion models (60.3%–90.5%) by up to +3.6%	Hybrid Fusion (HF-TIM), combining early fusion of BERT and VGG-19 features with late fusion through stacked unimodal predictions.	Hybrid fusion preserves unique unimodal features and complements them with multimodal integration, improving robustness and fine-grained classification.
An image and text-based multimodal model for detecting fake news in OSNs	[9]	2023	Text model: BERT with a dense layer. Image model: Xception (rescaled, Fakeddit-tuned) plus a visual sentiment polarity	Fakeddit	Acc, P, R, F1	Best result was with BERT+Dense and Xception using Maximum fusion, achieving 91.94% test accuracy with precision 93.76%, recall 92.83% and F1 93.29%	Maximum and Concatenate fusion	The multimodal fusion strategy reached around 93% F1, showing that combining textual and visual cues captures fake news patterns more effectively
SEMI-FND: Stacked ensemble based multimodal inferencing framework for faster fake news detection	[128]	2023	Stacked ensemble (BERT + ELECTRA), NASNet mobile	Twitter MediaEval and Weibo Corpus	Acc, P, R, F1, Parameter count	Twitter MediaEval dataset: 85.80% accuracy Weibo dataset: 86.83% accuracy	decision-level fusion: averaging	The suggested framework outperforms prior models like VGG-19 + TextCNNO
A Multimodal Fake News Detection Model based on Cross-image Semantic Fusion	[12]	2024	Text model: BERT. Image model: VGG-19	Weibo A, Weibo B	Acc, P, R, F1	The CISF model had an accuracy of 98.4%	Feature-level fusion: concatenation fusion	The proposed model outperformed other unimodal and multimodal models, however, has high computational power
Text–image multimodal fusion model for enhanced fake news detection	[85]	2024	Text encoder: BERT-Base. Image encoder: ResNet-50. For late fusion (LR, SVM, XGBoost, CatBoost)	Fakeddit, GossipCop	Acc, P, R, F1	Achieved 85% accuracy and 83% F1-score on GossipCop, and reached 90% accuracy with F1-scores 90% on Fakeddit	Early fusion: concatenation and element-wise addition. Joint fusion: element-wise multiplication with summation pooling. Late fusion: averaging	Combining text and image embeddings improves classification performance, with the joint + late fusion strategy offering the highest gains
Truth be told: a multimodal ensemble approach for enhanced fake news detection in textual and visual media	[22]	2025	Text model: SBERT+DeBERTa Image model:ResNet-50	Twitter MediaEval dataset, Wiebo dataset	Acc, P, R, F1	Accuracy of 87.4% on the Twitter dataset and an accuracy of 88.3% on the Wiebo NER dataset	Feature-level fusion: concatenation fusion	Although the model shows promising results, it has a high computational cost when compared to other multimodal approaches, and the gain in accuracy is small
Is it fake or not? A comprehensive approach for multimodal fake news detection	[11]	2025	Themis architecture	Fakeddit and ReCOVery	Acc, P, R, F1	On Fakeddit, MergeTokens + LoRA achieved 80.2% accuracy, and 97.5% on ReCOVery, TSIT + LoRA, outperforming existing models.	Feature-level fusion: Concatenation fusion	Multimodal fusion with targeted data augmentation and efficient fine-tuning significantly improves fake news detection, especially for imbalanced datasets

3.2.2 Attention-Based Fusion Approaches

While traditional fusion approaches are more common when constructing a multimodal architecture, some studies employ attention-based fusion strategies such as cross-modal attention [76]. This is a mechanism that enables one modality to focus on and extract meaningful information from another, allowing a computational model to flexibly weigh and integrate information across different data types, such as text and images, to capture deeper connections between them. Selected studies using this approach are provided in Table 3.4.

Yadav et al. [137] proposed an efficient transformer-based multilevel (ETMA) framework that jointly models text and images using three attention components: a visual attention-based encoder, a text attention-based encoder, and a joint attention stage that first applies visual-semantic attention to align important image regions with the most relevant words and then uses self-attention to filter redundant fused features. In this paper, Vision transformer (ViT) [68] was used to extract visual features, while BERT was used for textual features. The framework achieved 93% accuracy on the twitter dataset, 97% accuracy on the Jruvika dataset [138], 96% accuracy on the pontes dataset [139], and 95% accuracy on the Risdal dataset [140]. Overall, ETMA preserves modality-specific information while enforcing cross-modal alignment via attention, yielding consistent improvements.

Moreover, Shen et al. [141] introduced the Multimodal Contrastive Optimal Transport (MCOT) framework to detect fake news by combining text and image features. The model uses BERT to process text and ResNet to extract image features. It also applies cross-modal attention to connect the two modalities, contrastive learning to reduce the gap between text and image representations, and optimal transport to align their feature distributions. The framework was tested on the Weibo and Pheme [142] datasets and achieved higher accuracy and F1-score compared to existing models. Still, its dependence on pretrained models and limited dataset sizes may affect its ability to generalize to other domains.

In this direction, Tian et al. [143] introduced CMFNThinker, a cross-source multimodal model that integrates attention-based fusion with case-based reasoning for fake news detection. This model simulates the human thinking way used to protect fake news in three stages: starting with the perception stage using three models, BERT to extract domain-specific terms, with KeyBERT [144] to generate the keywords, and lastly a summarization model from the Fengshenbang 1.0 suite [145] to produce a condensed textual representations.

For case retrieval as the second stage, SBERT [133] is employed to measure the semantic similarity between new posts and those stored in the case library, while visual features are extracted using VGG-19. The final stage is used for reasoning, where cross-attention is applied to fuse textual and visual embeddings. Additionally, natural language inference is used to compare the retrieved cases with the target post, and to improved interpretability, social context features from similar post are merged. Experiments were conducted on Weibo-21, a single-source dataset used to train the model and serve as a case library, while MCFEND [146], a multi-source Chinese fake news dataset, was used for testing. The results clearly demonstrated that this strategy is highly effective, achieving a macro-F1 score of 0.902 on Weibo-21 and improving cross-source detection by more than 11% over state-of-the-art baselines. The study also found that both similar news retrieval and social context played an important role, making the model more accurate and interpretable for fake news detection across platforms.

Lu and Yao [147] proposed a high-performance multimodal fake news detection model that integrates residual convolutional networks with attention mechanisms. The model combines textual, visual, and even video data through a multimodal fusion framework. Specifically, it employs ResNet for image representation, BERT for text encoding, and a SlowFast network [148] for temporal video analysis. A weighted attention mechanism is introduced to align and fuse features across different modalities. Experiments on LIAR, FakeNewsNet, and Weibo datasets demonstrated superior results, achieving 97.7% accuracy and 92.4% F1-score, outperforming baseline models such as BERT, RoBERTa, XLNet, ERNIE, and GPT-3.5. The study highlights the effectiveness of attention-driven multimodal feature fusion in improving robustness, generalization, and efficiency for fake news detection tasks.

Additionally, Guo et al. [10] introduced the consistency-heterogeneity balanced multimodal framework (MFND-CMM) for fake news detection. The framework integrates textual, visual, and image-embedded text modalities, where BERT with self attention is applied to text and OCR-text, and an enhanced ResNet-50 with CBAM [77] extracts visual features. Cross-attention mechanisms are used to enable interaction among modalities, and pairwise similarity scores capture cross-modal consistency. The framework achieved 90.3% accuracy on the Weibo dataset and 93.2% on the Twitter dataset, outperforming 12 baseline models from both fusion-based and consistency-based categories.

Table 3.4: Summary of Attention-Based Multimodal Related Work

Article	Ref.	Year	Model/ Algorithm	Dataset	Metric used	Results	Fusion Strategy	Key Insight
ETMA: Efficient Transformer-Based Multilevel Attention Framework for Multimodal Fake News Detection.	[137]	2024	Vision transformer (ViT), BERT, Joint attention based learning: visual semantic attention and self attention	Twitter, Risdal, Jruvika, Pontes	Acc, P, R, F1, ROC AUC, and PR AUC	Twitter: 93% accuracy, Jruvika: 97% accuracy, Pontes: 96% accuracy, and Risdal: 95% accuracy	Feature level fusion with cross-attention and joint-attention	The computation time of the model is lower than the state-of-the-art methods.
Multimodal Fake News Detection with Contrastive Learning and Optimal Transport.	[141]	2024	MCOT framework with cross-modal attention, contrastive learning, and optimal transport classifier	Weibo and Pheme	Acc, P, R, F1	Achieved 90.1% accuracy and 90.3% F1-score on Weibo, and 87.0% accuracy and 77.9% F1-score on Pheme dataset.	Cross-modal attention with concate-nation of [CLS] token and pooled features	Relies on pretrained models and outperforms all baselines by integrating cross-modal attention, contrastive learning, and optimal transport.
CMFNThinker: A Cross-source Multi-modal Fake News Detection Model	[143]	2025	Three-stage framework with text summarization (BERT + KeyBERT + Fengshenbang), case retrieval (SBERT), and visual features (VGG19)	Trained on Weibo-21 and tested on MCFEND multi-source dataset	Acc, P, R, Macro F1, F1 decrease rate	Macro-F1 = 0.902 on Weibo-21. cross-source Macro-F1 = 0.580 (MCFEND-Group1) and 0.610 (MCFEND-Group2), outperforming baselines by 11.3%–13.2% and showing the lowest performance drop	Cross-modal fusion with reasoning over similar news and social context	The model mimics human reasoning using summarization, case retrieval, and inference; ablation shows similar news contributes most, and overall design improves interpretability and cross-source generalization
A Fake News Detection Model Using the Integration of Multimodal Attention Mechanism and Residual Convolutional Network	[147]	2025	Residual Convolutional Network (ResNet) + Multimodal Attention; integrates BERT (text), ResNet (image), and SlowFast (video) encoders	LIAR, Fak-eNewsNet, Weibo	AAcc, P, R, F1	Achieved 97.7% accuracy and 92.4% F1-score, outperforming baselines such as BERT, RoBERTa, XLNet, ERNIE, and GPT-3.5	Feature-level fusion: attention-based with residual connections	Combining multimodal attention and residual learning enhances robustness, scalability, and generalization across datasets.
Consistency-heterogeneity balanced fake news detection via cross-modal matching.	[10]	2025	Textual and OCR-text features: BERT with self-attention, Visual features: ResNet-50 + CBAM	Weibo dataset and Twitter dataset	Acc, P, R, F1	Weibo: 90.3% accuracy and Twitter: 93.2% accuracy. Outperformed 12 fusion-based and consistency-based baselines.	Cross-attention-based fusion	Introduced OCR-text as an auxiliary modality, Balanced consistency and heterogeneity, and Achieved significant improvements in accuracy across datasets.

3.3 General Lightweight Frameworks

Lightweight frameworks aim to reduce the complexity of detection models while maintaining high accuracy. Instead of relying on large networks with heavy computation, the studies below introduce lightweight frameworks that maintain a strong performance. This section discusses several areas of lightweight frameworks, including lightweight frameworks for deepfake detection task, unimodal lightweight frameworks for other similar tasks, and multimodal lightweight frameworks for other similar tasks.

3.3.1 Lightweight Frameworks for DeepFake Detection Task

In this section, two studies that focus on deepfake detection are presented, which targets manipulated facial videos rather than fake news. Both studies employ lightweight models which are designed to be more suitable for deployment on devices with limited computational resources. A summary of these studies is provided in Table 3.5.

Yasir and Kim [149] proposed a lightweight deepfake detection framework on feature-level concatenation fusion, where handcrafted descriptors are merged into a single combined feature vector. In this context, the term multi-feature fusion refers to the fusion strategy that concatenates Local Binary Pattern (LBP), Histogram of Oriented Gradients (HOG), and KAZE features before feeding them into the classifier. This design aims to address specific computational challenges associated with deep learning models, such as high GPU dependency, long training time, and large memory requirements that limit deployment on resource-constrained devices.The method integrates handcrafted feature descriptors, including LBP, HOG, and KAZE, which capture both texture inconsistencies and structural distortions in manipulated faces. These features are combined and evaluated using traditional machine learning classifiers such as Random Forest [150], XGBoost [116], Extra Trees [151], and Support Vector Classifier [113]. The model was trained and tested on benchmark datasets, namely FaceForensics++ [152] and Celeb-DF (v2) [153], achieving detection accuracies of 92% and 96%, respectively. This model is considered lightweight as it achieves high accuracy with reduced training and inference time compared to deep CNN-based approaches. Results demonstrate that the fused feature approach outperforms individual features provides a computationally efficient alternative to deep CNN-based models. The study highlights the complementary role of texture and keypoint based features in capturing subtle deepfake

artifacts, while suggesting that further improvements could enhance generalization to more diverse manipulation techniques.

Guo et al. [154] proposed TinyDF, a lightweight deepfake detection network designed to balance accuracy and efficiency. The model introduces a Pyramid Atrous Aggregation module for multi-perspective integration of local and global features and a Shuffle Fusion Mixer module for efficient cross-level feature interaction. In addition, a Kolmogorov Arnold Network was applied to improve nonlinear feature modeling. Evaluations demonstrated that TinyDF achieved 94.87% accuracy on FaceForensics++(c23) [152], 92.08% on Celeb-DF(v2) [153], 91.56% on WildDeepfake [155], 90.78% on Deepfake Detection Challenge (DFDC) [156], and 92.61% on DeeperForensics-1.0 [157]. These results show that TinyDF outperformed other lightweight methods, while requiring only 5.38M parameters and 0.59G FLOPs.

Table 3.5: Summary of Lightweight Frameworks For DeepFake Detection Task

Article	Ref.	Year	Model/ Algorithm	Dataset	Metric used	Results	Key Insight
Lightweight Deepfake Detection Based on Multi-Feature Fusion	[149]	2025	Multi-feature extraction (HOG for gradients, LBP for textures, KAZE for keypoints) + keyframe extraction (0.5s interval, resize to 28×28, grayscale)	FaceForensics++, Celeb-DF (v2)	Acc	92% accuracy on FaceForensics++, 96% on Celeb-DF(v2). Best performance achieved with HOG + KAZE fusion.	Fusion of handcrafted descriptors (HOG, LBP) with robust KAZE keypoints balances computational efficiency + high accuracy, making it suitable for lightweight, real-time deepfake detection on resource-constrained devices.
TinyDF: Tiny and Effective Model for Deepfake Detection	[154]	2025	TinyDF framework: Pyramid Atrous Aggregation, Shuffle Fusion Mixer, and Kolmogorov-Arnold Network	FaceForensics++, Celeb-DF-v2, WildDeepfake, Deepfake Detection Challenge, DeeperForensics-1.0	Acc, AUC, FLOPs	Accuracy: FaceForensics++: 96.7%, Celeb-DF v2: 90.0%, WildDeepfake: 80.6%, Deepfake Detection Challenge: 77.2%, DeeperForensics-1.0: 92.1%	Designed to balance accuracy and efficiency in deepfake detection, outperforms eight state-of-the-art models with higher accuracy and AUC while maintaining only 5.38M parameters and 0.59 GFLOPs.

3.3.2 Unimodal Lightweight Frameworks for Other Similar Tasks

This section reviews two studies that employ lightweight models for tasks beyond multimodal fake news detection, with one focusing on detecting object removal forgeries in surveillance videos [158], and the other on identifying pests and diseases in plants [159]. Table 3.6 provides a summary of these studies.

Sandhya and Kashyap [158] proposed an approach for detecting object removal forgeries in surveillance videos. Their method introduces a max averaging motion residue windowing technique to highlight temporal inconsistencies caused by object removal. The processed frames are then passed through a lightweight CNN that uses depthwise-separable layers for classification, which reduces the number of parameters to 1.24M, while maintaining higher accuracy. The proposed network achieved 98.6% frame-level accuracy and 99.01% video-level accuracy on the SYSU-OBJFORG dataset [160], which contains 200 HD videos (100 authentic and 100 forged).

The study in [159] proposed the Gated Asymmetrical GhostNet (GA-GhostNet), which is a lightweight CNN model used to identify pests and diseases in plants. The study utilizes four datasets, which are IP102 Pest [161], Jute Pest [162], Embrapa [163], and Apple Disease [164] dataset. In addition, the proposed model uses GhostNet [55] as its backbone architecture and replaces the SE and Ghost modules in GhostNet with Asymmetrical Ghost (AG) module and Gated Multi-Scale Coordinate Attention (GM -CA) module, respectively. Furthermore, the main building blocks of GA-GhostNet are the Gated Asymmetrical Ghost bottleneck (GAG-bneck) modules. The GAG-bneck contains two AG modules with the GM-CA module integrated between them. For the AG module, it first uses a 1×1 convolutional layer that reduces the number of input channels. Next, it utilizes the Asymmetric Group Convolution Block (AGCB) which consists of three branches. Each branch performs depthwise convolutions with with kernel sizes of 1×3, 3×3, and 3×1, respectively, to expand the feature maps. This will enhance feature extraction since separate branches can learn different directional features while keeping the computation cost low by concatenating the outputs into a single 3×3 convolution. Moreover, the GM-CA is an attention module that analyzes features at multiple scales to capture small details and large areas. It also specifies which features should be highlighted and what should be suppressed to prevent irrelevant details from affecting the final decision. Also, CutMix data augmentation and

transfer learning have been applied to enhance the model's performance. The results show that GA-GhostNet achieves a maximum accuracy of 71.9% on the IP102 dataset, while also achieving near-perfect accuracy on Jute, Embrapa, and Apple datasets outperforming existing lightweight models. Overall, GA-GhostNet presents a good balance between model size and performance. It reaches high accuracy on pest and disease datasets with about 3.73 million parameters using only around 168 million FLOPs. Models like MobileNetV2 have fewer parameters (around 2.35M) but much higher FLOPs (around 300M), and larger models like EfficientNet-B1 or MixNet-L, have many more parameters and FLOPs but without a better improvement in the accuracy. This demonstrates that GA-GhostNet is computationally efficient, delivering competitive or better performance with lower computational cost.

Table 3.6: Summary of Unimodal Lightweight Frameworks for Other Tasks

Article	Ref.	Year	Task	Model/ Algorithm	Dataset	Metric used	Results	Key Insight
A Light Weight Depthwise Separable Layer Optimized CNN Architecture for Object-Based Forgery Detection in Surveillance Videos	[158]	2024	Detecting object removal forgeries in surveillance videos	Depthwise separable layer optimized CNN with max averaging motion residue windowing	SYSU-OBJFORG	Acc, P, R, F1	Frame-level: 98.6% accuracy, video-level: 99.01% accuracy	High accuracy with reduced computational cost and parameters (1.24M) are achieved through lightweight depthwise separable CNN and motion residue features.
GA-GhostNet: A Lightweight CNN Model for Identifying Pests and Diseases Using a Gated Multi-Scale Coordinate Attention Mechanism	[159]	2024	Image-based identification of pests and diseases in plants	GhostNet as a backbone with replacing SE and Ghost modules with AG and GM-CA modules, respectively.	IP102 Pest, Jute Pest, Embrapa, and Apple Disease.	Acc, Macro-average P, Macro-average R, and Macro-average F1 score.	GA-GhostNet achieved an accuracy of 71.9% on IP102 Pest, 99.89% on Jute Pest, 96.97% on Embrapa Disease, and 95.17% on Apple Disease datasets.	GA-GhostNet efficiently balances model size and performance, achieving high accuracy with around 3.73M parameters and around 168M FLOPs, while other models have higher FLOPs or parameters without significant accuracy gain.

3.3.3 Multimodal Lightweight Frameworks for Other Similar Tasks

The studies below present lightweight multimodal frameworks in tasks outside the domain of multimodal fake news detection. They include tasks such as visual question answering in the medical domain [165] and multimodal gender and emotion recognition [166]. Additionally, some of these frameworks operate on different modalities, such as text, audio, videos, and images, and are not only limited to text and image pairs. A summary of these studies is provided in Table 3.7.

Liue et al. [166] proposed a lightweight multimodal framework for the multi-task of emotion and gender recognition using three modalities: audio, video, and text. They used the IEMOCAP [167] dataset, which contains videos, along with their audios and textual transcription. Additionally, they only used instances that have one of the four emotional labels (neutral, happy, angry, and sad). Since the dataset does not provide the gender label, the authors processed the dataset by adding a gender label based on the name associated with each instance. Next, features from each modality are extracted. Both audio and video data use MobileNet [13] as the feature extractor, where it outputs embeddings with dimensions of 512. However, audio data are first transformed into Mel frequency cepstral coefficient (MFCC) features before going through MobileNet to provide a better representation of human voice. The text data uses BERT [37] to extract features from the text, where it outputs embeddings with dimensions of 768. The study specifically focused on fine-tuning only the last three layers of BERT while keeping the rest of the layers frozen to maintain a lightweight design. Lastly, the output embeddings of all the modalities are fused using an attention based fusion mechanism. The results reveal that the proposed framework achieved an accuracy of 81.4% on the emotion recognition task and an accuracy of 83.6% on the gender recognition task while maintaining a parameter count of 8.4 million. Moreover, the study compared these results with an alternative architecture that replaces MobileNet with Resnet [46] . Although the architecture utilizing Resnet as the feature extractor obtained an accuracy of 83.8% on the emotion recognition task and 87% on the gender recognition task, it required 47.03 million parameters. This makes the model significantly larger while only offering a small gain in accuracy.

Within the scope of lightweight multimodal models, the Lite-MDETR study [168] provides a clear and practical example of how vision language systems can be made more

efficient while still performing well. The model's main task is to understand an image and a related text description together for example, identifying which object in the image the text is referring to, grounding phrases in specific regions, or answering a question based on both modalities. The authors evaluate these capabilities mainly on referring expression and phrase grounding datasets, including RefCOCO [169], RefCOCO+ [170], and RefCOCOg [171], which are widely used benchmarks for assessing region-level multimodal understanding. Traditional multimodal models rely on very large linear layers to combine image and text features, which makes them too heavy for real-world deployment. Lite-MDETR addresses this problem through the Dictionary Lookup Transformation (DLT), which replaces these heavy layers with a smaller dictionary and simple lookup-and-scale operations [168]. This allows the model to keep most of the knowledge learned by the original MDETR while avoiding the need for costly retraining. According to the results reported in the paper, this approach reduces the model size by up to four times and still achieves strong accuracy on tasks such as referring expression comprehension, phrase grounding, and visual question answering. Although a few larger models slightly outperform it in some settings, Lite-MDETR offers an excellent balance between efficiency and performance, requiring significantly less memory than typical multimodal transformers. In particular, the Lite-MDETR-TTQ variant is reported as the most compact configuration, with a model size of only 110MB.

Wang et al. [172] proposed LMFNet, a lightweight multimodal data fusion network developed for the task of multimodal semantic segmentation for remote sensing. The goal of this task is to classify each pixel in large satellite or aerial images into meaningful categories by using multiple types of image data such as RGB images, near infrared images, multispectral images, and depth maps, which usually requires large and heavy models to handle these different types of modalities. Therefore, the paper introduced the lightweight LMFNet, which is composed of, a weight-sharing multi-branch vision transformer backbone that extracts features from any type of image modality, and a multimodal fusion module, which consists of two main layers: a multimodal feature fusion reconstruction layer and multimodal feature self-attention fusion layer. As for the multimodal feature fusion refinement layer, which aligns and projects the extracted features into the same latent space to enable them to interact. Then, the multimodal feature self-attention fusion layer applies a self-attention mechanism to allow the aligned features from the different image modalities to interact and adaptively selects the most important parts, and then they are merged using a maximum

fusion operation. Finally, the fused features are passed through an MLP decoder to produce the final segmentation output. The results showed that the LMFNet framework achieved an mIoU of 85.09% on the US3D dataset [173] and 86.39% on the Potsdam dataset [174], and 82.49% on the Vaihingen dataset [175], while maintaining a small parameter count of 4.22 million.

Moreover, the study in [165] introduced a lightweight multimodal framework for visual question answering, specifically tailored to the medical field. The main task of the model is to take as input the image and the corresponding question, and an answer to the question is generated as output.The study integrated BioMedCLIP [176], which is a biomedical version of CLIP [177] pretrained on medical data, with LLaMA-3 [136], which is used to generate text. BioMedCLIP is the image encoder that reads both the image and the corresponding question. While LLaMa-3 is the text encoder that converts the input text into textual embeddings, fine-tuned using LoRA. Each of the encoders are trained separately on the OmniMedVQA dataset [178]. Additionally, some multiple-choice questions were modified by removing the multiple-choice options and replacing them with the truth, so that the model could accommodate both open-ended and closed-ended questions. Lastly, the embeddings from the two encoders were then fused within LLaMa-3 using feature-level fusion. The results show that the proposed architecture has a strong ability to handle open-ended questions, achieving an accuracy of 70.7%. However, in the case of closed-ended questions, some models outperform the proposed model, which achieved an accuracy of 76.9%, while the other models achieved up to 84.2% accuracy. Although some models showed a higher overall accuracy, the framework performs well considering it contains significantly fewer parameters, approximately 8 billion, compared to those existing in the field, which typically consists of 34 billion parameters. Therefore, it is more efficient, making it suitable for real-world implementation.

Table 3.7: Summary of Multimodal Lightweight Frameworks in Other Tasks

Article	Year	Task	Model/ Algorithm	Dataset	Metric used	Results	Fusion Strategy	Key Insight
A lightweight Multi-Modal Emotion Recognition Network Based on Multi-task Learning [166]	2021	Emotion and Gender Recognition using audio, video, and text	For audio and video encoding: MobileNet For text encoding: BERT	IEMOCAP	Acc, Parameters	Accuracy on Emotion Recognition: 81.4% , Accuracy on Gender Recognition: 83.6%, Total Parameters: 8.4 million.	Attention based Fusion	Only the last three layers of BERT were fine-tuned. Additionally, when compared to the version that replaces MobileNet with ResNet, the architecture that utilizes ResNet had slightly higher accuracy but is significantly larger, resulting in 47.03 million parameters.
Lite-MDETR: A Lightweight Multi-Modal Detector [168]	2022	Referring Expression Comprehension, Phrase Grounding, Referring Expression Segmentation, VQA	DLT (Dictionary Lookup Transformation) replacing heavy Linear Transform layers in MDETR; factorized and sparsified projection layers	RefCOCO, RefCOCOg, RefCOCO+, Flickr30k, PhraseCut, GQA	Acc, mIoU, Model size	Accuracy on Referring Expression (RefCOCO): 85.4%,Accuracy on RefCOCO+: 80.5%, Accuracy on RefCOCOg: 80.2%	No additional fusion, the model relies on MDETR's original cross-modal attention	Replaces large linear layers with dictionary-based lightweight projections; inherits pretrained knowledge without additional large-scale pretraining and achieves strong performance with significantly fewer parameters.
LMFNet: An Efficient Multimodal Fusion Approach for Semantic Segmentation in High-Resolution Remote Sensing [172]	2024	Remote Sensing Semantic Segmentation	Weight-sharing multi-branch Vision Transformer backbone with multimodal fusion module	US3D, Potsdam, Vaihingen	mIoU, Parameters	mIoU: 85.09% US3D, 86.39% Potsdam, 82.49% Vaihingen, Parameters: 4.22M	Novel multimodal fusion module	Achieves high performance with significantly fewer parameters than other multimodal methods, demonstrating excellent efficiency.
A Lightweight Large Vision-language Model for Multimodal Medical Images [165]	2025	Visual Question Answering in the Medical Field	For image encoding: BioMedCLIP For text encoding and generation: LLaMa-3 fine-tuned using LoRA	OmniMEDVQA	Acc, Parameters	Accuracy on open-ended questions: 70.7% Accuracy on closed ended questions: 76.9 %, Overall accuracy: 73.2%	Feature-level Fusion	Although having slightly lower accuracy, the proposed model is more cost-effective and resource-efficeint compared to other VQA models in the same field.

3.4 Applications of Siamese Network

This section highlights how Siamese Networks have been applied across various tasks that rely on learning similarity between paired inputs. As far as we are aware, they have not been previously applied to fake news detection, and therefore we present their use in related similarity-based tasks. Presenting these applications clarifies the role of Siamese architectures and demonstrates their effectiveness in tasks such as verification, matching, and tracking. Siamese networks operate by processing paired data through identical subnetworks that share weights, where each branch independently generates an embedding that is later compared using a distance or similarity function [179]. This design has enabled strong performance in similarity-based applications across domains, including object tracking [180]. A summary of related Siamese-based studies is presented in Table 3.8.

Jian et al. [75] proposed a Siamese Transformer Network to solve the few-shot image classification problem, a problem where images are classified based on small number of labeled images. In their method, two independent vision transformers (ViT) [68] were utilized, one on each branch, to extract different features from the input images. The features include the global features, which represents the overall features of the entire image, and the local features, which focuses on the features local to each region of the image. Each of the branches take two images as input; the support image, whose class is known, and the query image, whose label is unknown. The first branch is responsible for extracting the global features from these two inputs, whereas the second branch is dedicated to extracting the local features. Next, each of the output embeddings go through different similarity functions. The first branch leverages the Euclidean distance function, while the second branch uses Kullback-Leibler divergence to measure the similarity between the two input images. These two scores are then fused using an additive fusion strategy to obtain the overall score and determine whether these two images belong to the same class or not. In this study, multiple popular datasets were utilized to both train and test the model, including FC100 [181], miniImageNet [182]4, CIFAR-FS [183], and tieredImageNet [184]. The results show that the proposed model outperformed all other models that either only extract global or local features only across all datasets, achieving the best performance on the CIFAR-FS dataset with an accuracy of 90.81 % in the 5-shot 5-way setting.

The study in [185] proposed the SiamSMN framework for object tracking. In this study,

four datasets were utilized which are COCO [186], ImageNet VID [187], ImageNet DET [187], and LaSOT [188]. The framework of SiamSMN consists of three components which are a feature extraction network, a multi-scale fusion module, and a prediction head. The feature extraction network itself contains two subnetworks. Furthermore, ResNet-50 was used as the backbone network for both subnetworks which share the same parameters. At first, the template image and the search image were individually processed by their corresponding subnetworks to extract features. The resulting feature maps were then projected into a shared embedding space, where cross correlation is applied to produce similarity maps. Cross correlation combines each pair of feature maps, one from the template and the other from the search, to measure their similarity. Then, the resulting similarity maps were used as input into the multi-scale fusion module which consists of an encoder and a decoder. The encoder aims to learn how to fuse multiple similarity maps and to analyze the relationships and dependencies among them. This will produce a fused, meaningful map that enters as input to the decoder. The decoder takes the fused map from the encoder and refines it to produce a final response map. It combines high-level and low-level features to improve tracking accuracy and locate the target. Moreover, the prediction head consists of three branches which are classification, regression, and centerness. These branches work together to locate the target and estimate its size and shape. In this study, a box refinement module was used to adjust the bounding boxes which takes the regression output, feature map, and response map as inputs and adjusts the box boundaries to produce more accurate results.

Additionally, Huertas-Tato et al. [189] proposed SimCLIP, which is a Siamese-based multimodal model designed for meme classification. This task aims to classify the content of a meme into humor, offensiveness, harmful content, or sarcasm. The model depends on CLIP, a vision-language framework that uses contrastive learning across a large number of image and text pairs, and its main advantage is that it aligns visual and textual embeddings in the same latent feature space [177]. In SimCLIP, the authors use a pretrained CLIP encoder and freeze it to extract image and text embeddings. They freeze it to keep both modalities aligned in the same shared embedding space. In this approach, the text is extracted from the meme image using OCR, then both image and text are processed through CLIP encoders, and their embeddings are then projected using lightweight feed-forward layers. In this paper, the authors refer to their fusion as a Siamese fusion strategy, where the two embeddings are combined using concatenation, absolute difference, and Hadamard product to capture

both semantic agreement and contradiction between modalities. At the end, the fused features are used by the task classifier to make the prediction. The model was evaluated on several meme datasets, including Memotion7k [190], FBHM [191], Harm-C and Harm-P [192], MultiOFF [193], and Propaganda [194]. The results showed that SimCLIP achieved strong performance, obtaining state-of-the-art results on Memotion7k Task 3 and F1-scores that surpass human performance on Harm-P.

Ye et al. [195] introduced a novel Multimodal Features Alignment (MFA) framework for vision language object tracking, a task that aims to locate and follow an object in a video using a natural-language description rather than relying solely on visual cues [195]. The proposed model enhances the interaction between textual and visual modalities within a Siamese-based network. It employs cross-modal bilinear fusion, a technique that multiplies the text and image feature dimensions to capture higher-order cross-modal relationships [196], together with a dual co-attention mechanism, which enables both modalities to attend to the most informative regions of each other and strengthen semantic alignment [91]. The fused features serve as the search region for the Siamese tracker, replacing standard resized inputs and enabling more accurate target localization. The model achieved results on OTB-LANG [197] , LaSOT, and TNL2K [198] and showed strong results. These findings confirm that combining semantic alignment with attention-based fusion improves feature representation and tracking accuracy.

Wan et al. [199] proposed Sigma, a siamese network for the task of multimodal semantic segmentation, which uses RGB images with other image modalities to assign a semantic label for each pixel, which divides the image into meaningful regions or objects. To accomplish this task, Sigma integrates several components from Mamba [200] and VMamba [201] such as Visual State Space Block, Cross Mamba Block, and Concat Mamba Block. Mamba is state space model used for sequence modeling tasks such as language modeling and time-series forecasting, it is competitive with transformers while replacing attention mechanisms with a selective state space mechanism, it has a fast inference speed and is computationally efficient. VMamba is derived from Mamba but focuses on visual modalities by integrating visual state space blocks, which help with spatial dependencies in images. The paper uses RGB images as a main modality and for the second modality either thermal or depth images can be used, as the model was trained for both modalities. The network consists of three

main parts, a siamese encoder, a fusion module, and a decoder. For each encoder branch, one image type is taken as input, and features are extracted using Visual State Space Blocks to capture important information. The fusion module consists of two main parts which combine the features from both branches. The first part is the Cross Mamba Block, which allows the two modalities to exchange information and to learn how the other modality maps its outputs. The second part is the Concat Mamba Block, which concatenates the two representations together to form a single feature map. Finally, the decoder then uses a Channel-Aware Visual State Space Block to highlight the important information in the feature map and produce the final segmentation output. The Sigma network was trained and tested on multiple datasets including MFNet [202], PST900 [203], NYU Depth V2 [204], and SUN RGB-D [205], which include RGB, depth, and thermal images. As for results, Sigma outperformed several baseline models on the MFNetand the PST900 datasets.

Table 3.8: Summary of Siamese Network Applications

Article	Year	Task	Model/ Algorithm	Dataset	Metric used	Results	Fusion Strategy	Key Insight
Siamese Transformer Networks for Few-Shot Image Classification [75]	2024	Few-Shot Image Classification	Siamese Transformer Network containing ViT on both branches	FC100, miniImageNet, CIFAR-FS, tieredImageNet	ACC	90.81% on CIFAR-FS, 90.71% on tieredImageNet, 88% on miniImageNet, 66.9% on FC100	Additive fusion strategy	This papers' approach outperformed all others in the few-shot image classification problem. This paper also leveraged two different distance functions to compute similarity.
SiamSMN: Siamese Cross-Modality Fusion Network for Object Tracking [185]	2024	Object tracking	ResNet-50 as the backbone network	COCO, ImageNet VID, ImageNet DET, and LaSOT	AUC, Average Overlap (AO), Success Rate (SR)	SiamSMN achieves 72.7% on OTB100, 66.0% on UAV123, 63.2% on GOT-10K, and 64.7% on LaSOT.	Multi-scale fusion module which consists of an encoder and a decoder	Multi-scale fusion that combines spatial details and semantic information is more effective than traditional methods, since it results in higher tracking accuracy.
A CLIP-Based Siamese Approach for Meme Classification [189]	2024	Meme classification: humor, offensiveness, sarcasm, or harmful content	CLIP-based Siamese fusion network	Memotion7k, FBHM, Harm-C, Harm-P, MultiOFF, Propaganda	Macro F1, ACC, AUROC	Achieved State-of-art on Memotion7k Task 3 and super-human F1-score +13.7% on Harm-P	Siamese fusion strategy (concatenation, absolute difference, Hadamard product)	The Siamese fusion strategy provided the best performance among fusion methods while remaining lightweight.
Multimodal Features Alignment for Vision–Language Object Tracking [195]	2024	Vision–Language Object Tracking	Siamese-based Multimodal Features Alignment (MFA) network	OTB-LANG, LaSOT, TNL2K	AUC, P, P_{norm}, Fps	Achieved the best performance across OTB-LANG, LaSOT, and TNL2K benchmarks with balanced accuracy and real-time speed (37 Fps).	Cross-modal bilinear fusion with dual co-attention	Introduced semantic alignment and attention-guided fusion to improve multimodal tracking accuracy.
Sigma: Siamese Mamba Network for Multi-Modal Semantic Segmentation [199]	2025	Multimodal semantic segmentation	Siamese Mamba Network	MFNet, NYU Depth V2, PST900, and SUN RGB-D	mIoU	61.3% mIoU on the MFNet dataset and 88.6% mIoU on the PST900 dataset	Cross Mamba Block and Concat Mamba Block	Combines features from different image modalities using a Siamese design and Mamba-based fusion to improve segmentation accuracy and efficiency.

3.5 Discussion

This section discusses the findings from the reviewed studies and relates them to the goal of this research. It highlights the strengths and weaknesses of the existing methods used for fake news detection. The discussion starts with unimodal approaches, which focus on either text or images alone, and then transitions into multimodal approaches that integrate both.

Unimodal approaches are effective for detecting fake news based on one modality, such as text or images. Text-based models that use deep learning, such as LSTM and Transformers in [3, 108], can learn how words and meanings relate, helping them spot fake or misleading text. Likewise, image-based models such as CNNs can detect signs of manipulation in images, achieving high accuracy when fake content is only visual [6, 7]. This suggests that unimodal systems are effective mainly when the detection is limited to text or images, as shown in several prior studies [3, 6, 114].

However, these approaches still face important limitations. Because they analyze only one modality, they cannot understand the full meaning of news that combines both text and images. They often perform well on specific datasets but fail to generalize well to multimodal data. Furthermore, with the rise of digital media, the most prevalent form of news spread, whether fake or real, are those that combine text and images. This makes unimodal systems less effective, as they cannot capture the interaction between the two modalities. Therefore, developing multimodal models that integrate text and visual data is essential. As a result, these models can effectively handle complex and realistic fake news, improving their performance in real-world scenarios.

The multimodal approach for fake news detection achieves higher accuracy compared to the unimodal approach. The reason for this improvement is that, in most cases, fake news involves more than one modality. Text features focus on the words and the writing style, while visual features focus on what the image shows and its context. However, when these two modalities are fused together, they capture more information, which allows the model to detect fake news more effectively. For example, [8] achieved 87% accuracy with a multimodal approach compared to 78% with text-only BERT using the Fakeddit dataset, while [9] reached 91.94% accuracy by combining BERT and Xception features.

Different fusion strategies play a critical role in determining the model's performance and

accuracy. Many studies used feature-level concatenation because it is simple, computationally efficient, and effective at preserving the complete information from both modalities [8, 12, 22, 11, 10]. This paper [85], introduced many other feature-level fusion strategies, like element-wise addition, element-wise multiplication and summation pooling. However, it was shown in [9] that the model achieved higher accuracy using a feature-level maximum fusion strategy compared to concatenation. Similarly, [128, 85] applied an averaging-based decision-level fusion method that also produced competitive results. Moreover, [126, 85] have implemented hybrid fusion to improve the model's performance since it captures cross-modal interactions from early fusion and preserves modality-specific predictions from the late fusion.

Recent studies have also explored attention-based fusion mechanisms to enhance how text and image modalities interact. For instance, [141] used cross-modal attention which improved alignment between the two modalities, while [137] incorporated joint-attention to preserve the information of each modality while capturing cross-modal relationships. Similarly, [143, 147] integrated cross-attention and weighted attention mechanisms to refine how textual and visual features are aligned and combined.

Building on the fusion strategies discussed above, recent studies [189, 195] have explored Siamese-style fusion strategies, where the model compares image and text embeddings using operations such as concatenation, absolute difference, and Hadamard product to highlight both semantic similarity and contradiction between modalities. In SimCLIP [189], this strategy enabled the model to detect misleading cases where the caption does not align with the image.

These findings highlight the importance of exploring a wider range of fusion strategies, as different techniques may be more effective depending on the dataset and model architecture.

Furthermore, most of the reviewed studies used the Fakeddit dataset [8, 9, 126, 10], showing it has become a common baseline dataset for evaluating multimodal approaches. Many papers use Fakeddit to show the power of multimodal approaches over text-only models. Similarly, [9] used the Fakeddit data set to test different fusion strategies. Therefore, Fakeddit provides a reliable benchmark for testing model architectures and comparing different fusion strategies. In addition, the Twitter MediaEval dataset [24] has also been widely used as a standard benchmark in several studies [128, 22, 125] due to its focus on real world

events and its structure of pairing text with the corresponding image, making it suitable for multimodal fake news detection.

It is also important to note that images employed in multimodal fake news detection are not always manipulated, they may be genuine. It is common in datasets such as Fakeddit [23] and Twitter MediaEval [24] for real images to be paired with a false or out-of-context textual caption. In Fakeddit, one of the 6-way classification labels of the dataset is "False Connection", which describes samples where the text does not support the image. These samples are labeled as fake in the 2-way classification. Additionally, the inconsistent pairs in Twitter MediaEval are classified as fake. Therefore, previous studies rely on fusion strategies to capture cross-modal relationships and inconsistencies between text and image pairs, as done in [8, 126, 22].

In recent studies, there has been a clear move from traditional CNN-based models to transformer and parameter-efficient models. Several studies combined BERT for textual representation with CNN-based architectures such as ResNet or VGG-19 for images, as seen in [126, 12, 10]. Similarly, paper [22] combined SBERT and DeBERTa transformers with ResNet-50 to improve how text and image features are understood together. Likewise, [11] used LoRA to make the model faster and less expensive to train while keeping high accuracy. Both [11, 137] integrated ViT for image representation alongside BERT for text, reflecting a shift towards transformer architectures. This shows that researchers are now focusing not only on improving results but also on making multimodal fake news detection models more efficient and practical to use. Also, the papers show a frequent pattern of using transformer BERT for textual feature extraction.

Moreover, several studies [166, 165, 172, 168] have implemented a lightweight multimodal framework in tasks beyond multimodal fake news detection. The study in [166] proposed a lightweight architecture for the multi-task of emotion and gender recognition. The framework consists of MobileNet as the audio and video encoder, and BERT, with its last three layers fine-tuned, as the text encoder, resulting in a total parameter count of 8.7 million. When compared to an alternative architecture that consists of ResNet rather than MobileNet, the accuracy increased only slightly while the framework became significantly heavier, with a total of 47.03 million parameters. This demonstrates that utilizing lightweight models, such as MobileNet, can achieve competitive performance, while maintaining a small model size,

supporting our choice to adopt similar lightweight models in our architecture.

Additionally, the study in [165] claimed that their visual question answering multimodal framework is lightweight, however, it is considered lightweight in the medical domain of visual question answering and does not extend to our domain of multimodal fake news detection. This is because the framework mentioned in the study is relatively computationally heavy, containing 8 billion parameters and requiring extensive resources.

Although the review shows significant advancements in multimodal fake news detection, such as the use of CLIP and ViT in [11] to achieve high accuracy, and the development of cross-image semantic fusion in [12] to process several images as input rather than relying on a single image, most of these approaches are computationally heavy, requiring extensive resources. Based on these observations, the review shows a clear limitation of a lightweight multimodal framework in the domain of fake news detection that efficiently and accurately detects fake news items. Therefore, in our work, we aim to employ lightweight models and explore different fusion strategies that will effectively analyze multimodal fake news composed of both image and text data, while remaining efficient and maintaining high accuracy. Accordingly, we will answer the following research questions:

- RQ1: "Can a lightweight architecture be effectively introduced for multimodal fake news detection while achieving competitive performance?"
- RQ2: "How do different fusion strategies affect the overall performance of multimodal fake news detection approaches?"

3.5 Summary

In this chapter, a review was conducted on the different approaches towards fake news detection. Unimodal approaches were first explored, focusing mainly on text data, with limited work done on image data. Then, multimodal approaches that integrate image and text data were investigated, categorized based on their fusion strategies: traditional fusion and attention-based fusion. Additionally, some lightweight frameworks that aim to reduce the complexity of the models while maintaining high accuracy were examined. Different applications of Siamese Network are also presented. Lastly, a discussion was conducted to highlight the main findings in the reviewed study.

Chapter 4: Proposed Architecture

This chapter provides a detailed description of the two proposed architectures: the baseline architecture and the parameter-efficient cross-modal attention architecture. It describes the text and image encoders used in both architectures, the projection heads that map the embeddings or tokens into a common dimensional space, and the fusion strategies employed in each. Additionally, the chapter discusses the similarity measurement branch, which is applied only in the baseline architecture, and describes the classification layer that is used to generate the final prediction.

4.1 Design Overview

Since fake news often appears in multiple modalities, we propose a lightweight multimodal framework that integrates both text and visual modalities. The integration of these modalities helps overcome the weaknesses of single modality approaches [3, 6], such as using text or images alone, in detecting fake news. Moreover, unlike most existing multimodal fake news detection approaches that are computationally heavy [11, 12], our framework is designed to be lightweight. In addition, we will explore different fusion strategies, since the way text and images are fused can greatly affect the model's performance and also implement a similarity measurement branch to capture cross-modal alignment.

Our proposed framework employs MobileNetV3 [25] as the image encoder and TinyBERT [15] as the text encoder. The main reason for selecting these two specific encoders is that they are designed to be lightweight, which aligns with our goal. Moreover, they have a good trade-off between accuracy and computational efficiency. A detailed explanation of the encoder selection, along with the experimental protocols and findings for both the image and text encoders is provided in Chapter 5.

An overview of the proposed framework is provided in Figure 4.1. It consists of five components: the image and text encoders, the projection head, the fusion operation, the similarity measurement function, and the classification layer. To begin, the text and its corresponding image are each fed into their respective encoders for feature extraction. The main purpose of these encoders is to transform the raw data into meaningful feature representations that

can be effectively used. After feature extraction, the feature vectors from the text and image encoders are each passed through their own projection layer. The projection heads maps both vectors to the same dimensions, allowing them to be compatible. A similarity function is then applied to the projected image and text feature vectors during training to asses their alignment.Moreover, following projection, the feature vectors are fused using fusion strategies, and the resulting representation is passed through a classification layer to produce the final prediction. Each component is explained in detail in the following sections.

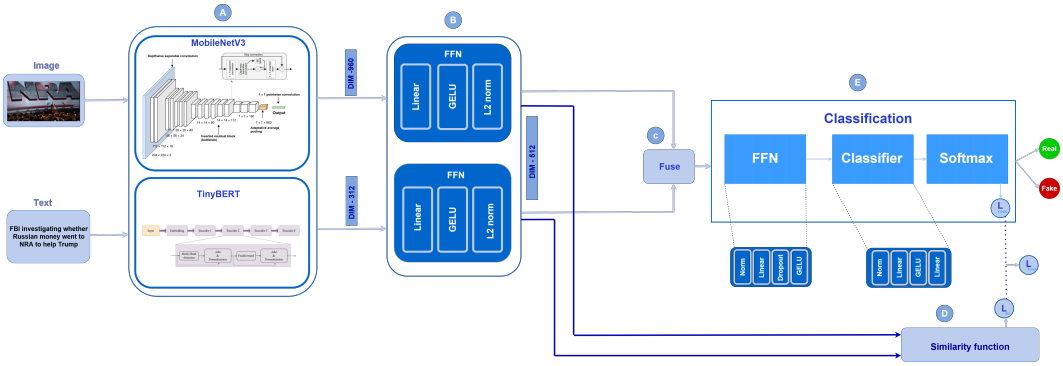


Figure 4.1: Overall architecture of the proposed multimodal model based on feature-level fusion. **(A)** Image and text encoders. **(B)** Projection head. **(C)** Fusion module. **(D)** Similarity function. **(E)** Classification layer.

4.2 Selected Image Encoder: MobileNetV3-Large

MobileNetV3-Large [25] is introduced as the next generation of MobileNet family that focuses on improving the balance between accuracy and speed, especially when running on mobile or edge devices. It is specifically tuned for mobile CPUs, therefore it can perform well without needing strong hardware. The model is designed using platform-aware Neural Architecture Search (NAS) [206] paired with NetAdapt [207], which helps adjust the network layer by layer. After that, several manual architecture improvements were added to further enhance performance. As a result, MobileNetV3-Large provides higher accuracy and lower latency than MobileNetV2, although this comes with an increase in the number of parameters [25].

MobileNetV3-Large is built on the same inverted residual block structure introduced in

MobileNetV2, but it adds two important improvements: the Squeeze-and-Excitation (SE) [53] module and the hard-swish activation function, to overcome the limitation of sigmoid computational cost on mobile hardware, and the loss of negative numbers caused by ReLU.

As illustrated in Figure 4.3, the architecture starts by receiving as input an RGB image of size $224^2 \times 3$, the network first applies a 3×3 convolution with stride 2, producing $112^2 \times 16$ features with the use of hard-swish nonlinearity. Intuitively, this layer learns basic edge and texture filters while downsampling the image by 2 to cut computation early. The hard-swish as shown in Equation (10), where the constants 3 and 6 follow directly from the design proposed in the MobileNetV3 paper.

$$h - swish(x) = x \cdot \frac{ReLU6(x+3)}{6} \tag{10}$$

Next, the output goes into a sequence of inverted residual bottleneck blocks (bneck). Each bneck follows the structure introduced in MobileNetV2: a 1×1 expansion conv increases the number of channels, followed by a depthwise conv (3×3 or 5×5 kernels) to perform spatial filtering channel by channel, and then a 1×1 projection convolution that reduces the channels back to the block's output size. When the input and output dimensions are the same and stride equals 1, a skip connection (residual) is used to preserve the original information. The use of the SE is applied in several mid and late blocks where the feature maps are more semantically informative. SE scale each channel by applying global average pooling followed by two small fully connected layers and the sigmoid function was replaced with its piece-wise linear hard analog as shown in Equation (11).

$$h - sigmoid(x) = \frac{ReLU6(x+3)}{6} \tag{11}$$

After the SE step, the network adjusts the importance of each channel so that more useful features are strengthened while less relevant ones are reduced. The activation function of the architecture changes as the network goes deeper. Early bottleneck blocks use ReLU, while the later blocks use the hard-swish activation to improve accuracy without adding extra computational cost. This design keeps the model efficient by maintaining small input and output channel sizes, while allowing richer feature learning inside the expanded intermediate layers.

The second bneck is where the first spatial downsampling occurs in the bneck sequence. Starting from $112^2 \times 16$, the block expands the channels to 64, applies a 3×3 depthwise conv with stride 2 to reduce the resolution, and then projects to 24 channels, resulting in $56^2 \times 24$. The following bottleneck keeps this resolution and expand the size to 72, reapply depthwise 3×3, with stride 1, and project back to 24 producing $56^2 \times 24$ again with a residual connection. Both of these early blocks use ReLU and do not include SE, focusing on building fast low-level features efficiently.

The next group of bneck layers increases spatial context and begins using SE. The first three switches to a 5×5 depthwise kernel and downsamples. $56^2 \times 24$ is expanded to 72 channels, filtered, and projected to 40 channels, resulting in $28^2 \times 40$. The next two follow the same resolution, each with stride 1 expanding the channels to 120, then projecting back to 40. These wider kernels and channel reweighting help capture richer mid-level information while remaining efficient due to depthwise computation. The larger 5×5 kernels and SE weighting help to capture richer mid-level information.

At this depth of the network, ReLU is no longer used, instead the hard-swish activation is applied for the remaining layers. The next bneck expands the channels to 240, applies a 3×3 depthwise conv. with stride 2, and then projects to 80 channels, reducing the spatial resolution to $14^2 \times 80$. The following three bneck blocks keep the same spatial resolution while only using different expansion sizes to 200 or 184 channels before projecting back to 80.

In the final set of bneck layers, the SE module appears again together with larger expansion sizes and the use of hard-swish. At $14^2 \times 80$, the block expands the channels to 480, applies a depthwise 3×3 conv, and then projects to 112 channels. This is followed by a second block with the same structure. After that, a 5×5 depthwise conv with stride 2 reduces the resolution to 7^2, while expanding to 672 channels and projecting to 160. Two more bneck blocks operate at this 7^2 resolution, each expanding to 160 and projecting back to 160 with stride 1, resulting in $7^2 \times 160$ compact of high-level features that form the final representation before classification.

Finally, a 1×1 convolution increases the number of channels to 960, followed by a 7×7 global average pooling which compresses the spatial dimensions to 1×1. A final 1×1 convolution expands the features to 1280 with hard-swish activation, and the last projection

layer maps this representation to the number of output classes.

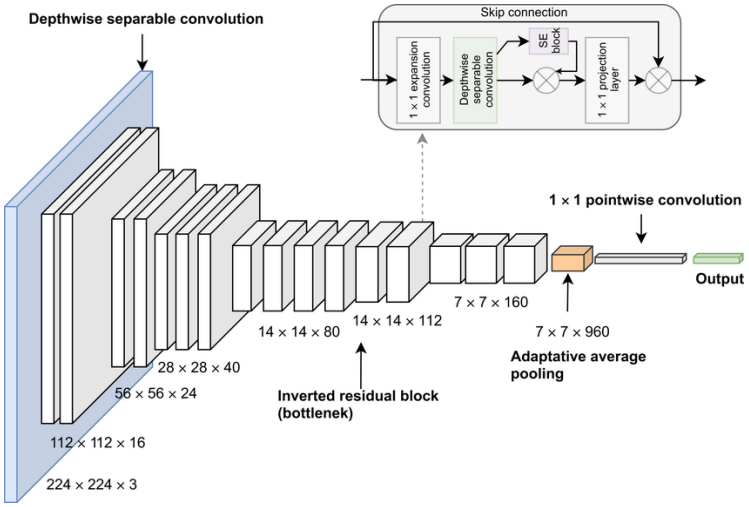


Figure 4.3: MobileNetV3 architecture with its core components [208].

4.3 Selected Text Encoder: TinyBERT

TinyBERT [15] is a smaller and more efficient version of BERT. It distills knowledge from BERT during training to achieve similar performance while reducing the number and size of Transformer layers. The model consists of an embedding layer followed by 4 encoder layers, which are stacked on top of each other.

TinyBERT first starts with the same embedding layer used in BERT, which combines token embeddings, positional embeddings, and segment embeddings, and each embedding is represented as a 312 dimensional vector. For every token, these three vectors are added together, then normalized, and finally passed through a dropout layer. This process results in a matrix that has a size of $L \times 312$, where L is the token sequence length.

The output of the embedding layer is passed as input to the stack of four Transformer encoder layers. Each encoder layer uses the same structure used in BERT and consists of two main components: a multi-head self-attention mechanism and a feed-forward network. The structure and dimensionality of the encoder is the same across all 4 layers.

The self-attention mechanism used in each encoder layer allows every token to consider and

pay attention to the other tokens in the sequence when creating its representation, allowing the model to capture more complex relationships between words. The attention is calculated as shown in Equation (12):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{12}$$

where Q, K, and V are the query, key, and value matrices, and d_k is the dimension of each attention head. In TinyBERT, the hidden size of 312 is divided across 12 attention heads, which allows the model to perform the attention mechanism in parallel making it faster. Next, the results from all the heads are then concatenated and projected back to 312 dimensions through a linear transformation. Afterward, a residual connection is added between the input and the output of the attention sublayer, and normalization is applied to stabilize the values and improve learning.

After the attention step, the output of the attention sublayer is passed into a feed-forward network, which contains two fully connected layers. The first layer expands the dimensions from 312 to 1200 and applies the GELU activation function [209], which is shown in Equation (13), while the second layer reduces the dimensions back to 312.

$$\text{GELU}(x) \approx 0.5x\left(1 + \tanh\left[\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right]\right) \tag{13}$$

Additionally, a second residual connection is then applied between the input and the output of the feed-forward network, and normalization is applied. Finally, this process completes one encoder layer, the same process is repeated across the rest of the encoder layers, with each one taking as input the output of the previous layer.

In conclusion, TinyBERT maintains the same architecture as BERT but with a reduced size. By using 4 layers, a hidden size of 312, and 12 attention heads, it manages to be lighter and faster. The TinyBERT architecture is illustrated in Figure 4.4.

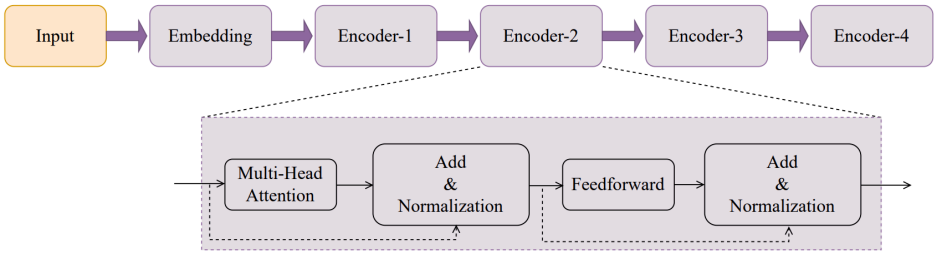


Figure 4.4: TinyBERT architecture.

4.4 Modality Embeddings

This section describes how the visual and textual information is represented before entering the fusion stage. Encoders are used to extract embeddings from each modality rather than using them for classification.

For the image branch, MobileNetV3 is used as the feature extractor. The last expansion layer and the classification head are removed, and the embeddings are extracted after global average pooling, resulting in a 960-dimensional feature vector. This embedding represents the image semantic information, which would be used as the final compact representation of the image.

As for the text branch, TinyBERT will provide the extracted features, where the version without a classification layer is used. From the last encoder layer, the [CLS] token embedding with 312 dimensions is extracted, which represents the entire input token sequence and is used as the final text representation.

4.5 Projection Head

Projection head is a shallow feedforward neural network that is commonly used to map the output features of an encoder into a low-dimensional space [210]. In our project, projection heads are needed to map embeddings from the MobileNetV3 encoder and the TinyBERT encoder to the same dimensions, allowing them to be compatible and efficiently fused together. We employed the projection heads similar to the one in study [189], where each

encoders have their own projection head with the same architecture, as shown in Figure 4.1 (B). The structure of the projection head starts with a single linear layer, followed by a GELU activation function that adds non-linearity.

Lastly, L2 normalization, as shown in (14), where e is the input vector and ϵ is a small positive value, is applied to normalize the embeddings [211].

$$\hat{\mathbf{e}} = \frac{\mathbf{e}}{\sqrt{\sum_{i=1}^{d} e_i^2 + \epsilon}} \tag{14}$$

The TinyBERT and MobileNetV3 encoders produce embeddings of 312 and 960 dimensions, respectively. Therefore, we mapped their embeddings to the same dimension of 512 to obtain balanced and compatible vectors.

4.6 Fusion Strategies

Fusion plays a key role in multimodal architectures, as it determines how text and image representations are combined into a single vector, as illustrated in Figure 4.1 (C). In this project, we will explore and compare four feature-level fusion strategies to examine how different fusion mechanisms affect multimodal representation quality and overall model performance.

(1) Concatenation Fusion: Concatenation fusion directly connects the text and image embeddings along the feature dimension to form a joint representation, preserving modality-specific information without alteration [82]. Several prior studies [8, 9, 10, 11, 12] have adopted concatenation-based fusion due to its simplicity and effectiveness. The concatenation fusion process can be mathematically expressed as (15), resulting in a 1024-dimensional vector.

$$F_{\text{fused}} = \text{Concat}(F_{\text{text}}, F_{\text{image}}) \tag{15}$$

(2) Element-wise Maximum Fusion: Element-wise maximum fusion selects for each feature dimension the larger value between the aligned text and image embeddings, emphasizing the strongest and most informative signals across modalities [212] and resulting in a

512-dimensional vector, as defined in (16).

$$F_{\text{fused}}^{(i)} = \max\left(F_{\text{text}}^{(i)}, F_{\text{image}}^{(i)}\right) \tag{16}$$

(3) Siamese-style Fusion: As referred to in the paper [189], the embeddings are combined using concatenation, absolute difference, and Hadamard product operations, as shown in Equation (17), which will result in a 2048-dimensional fused vector. This operation may allow the model to capture both semantic agreement and contradiction between modalities.

$$E_{\text{fused}} = [(E_{\text{TXT}}, E_{\text{IMG}}), |E_{\text{TXT}} - E_{\text{IMG}}|, E_{\text{TXT}} \odot E_{\text{IMG}}] \tag{17}$$

(4) Hadamard Product with Element-wise Summation Fusion: The two embeddings are combined first by applying an element-wise product, as shown in Equation (18). This gives an interaction vector which then will be element-wise summed with the original embeddings, as presented in Equation (19), allowing the model to integrate unimodal features with cross-modal interactions. This will produce a 512-dimensional fused vector.

$$E_{\text{interact}} = E_{\text{TXT}} \odot E_{\text{IMG}} \tag{18}$$

$$E_{\text{fused}} = E_{\text{TXT}} + E_{\text{IMG}} + E_{\text{interact}} \tag{19}$$

4.7 Similarity Measurement branch

To aid in the classification of fake and real news, measuring the similarity between the text and image feature representations may provide valuable cues about cross-modal consistency [213]. Inconsistency between image-text pairs often implies that the pair is fake, whereas real news typically exhibit stronger alignment between modalities. Therefore, based on this observation, a similarity measurement branch is introduced, as in [213, 141] , which operates on the projected text and image vectors. This branch computes an alignment loss using matched and mismatched image-text labels rather than fake and real labels. In this project, two similarity-based loss functions are evaluated to assess their effectiveness in capturing multimodal consistency, as illustrated in 4.1 (D).

(1) Cosine Similarity Loss: To compute the cosine similarity loss, let e_t and e_v represent the projected text and image embeddings, respectively. The cosine similarity between the two modalities is computed, as shown in (20), where $s \in [-1, 1]$ denotes the similarity score, and higher values of s indicate a stronger alignment between the text and image pairs [213].

$$s = \frac{e_t \cdot e_v}{\|e_t\| \, \|e_v\|\}}, \tag{20}$$

The similarity score s is then mapped into a probability using the sigmoid function, as illustrated in (21).

$$p_s = \frac{1}{1 + e^{-s}} \tag{21}$$

Lastly, given a matched or mismatched label y, the cosine similarity loss, denoted by L_s, is defined using the binary cross entropy loss, as shown in Equation (22).

$$\mathcal{L}_s = -\left[y \log(p_s) + (1 - y) \log(1 - p_s)\right] \tag{22}$$

(2) Contrastive loss: To compute the contrastive loss, the projected text and image embeddings e_t and e_v are considered. The similarity scores are computed using the dot product between the two vectors, which is scaled by the temperature parameter τ, as shown in (23). This results in a similarity matrix for text-to-image, denoted as $p_{t \to i}$, and another for image-to-text, denoted as $p_{i \to t}$.

$$\begin{aligned} p_{ij}^{t \to i} &= \frac{\exp\left(\mathbf{e}_t^i \cdot \mathbf{e}_v^j / \tau\right)}{\sum_{k=1}^{N} \exp\left(\mathbf{e}_t^i \cdot \mathbf{e}_v^k / \tau\right)} \\ p_{ij}^{i \to t} &= \frac{\exp\left(\mathbf{e}_v^i \cdot \mathbf{e}_t^j / \tau\right)}{\sum_{k=1}^{N} \exp\left(\mathbf{e}_v^i \cdot \mathbf{e}_t^k / \tau\right)} \end{aligned} \tag{23}$$

Then, the contrastive loss is computed by calculating the difference between the predicted similarity scores and the actual labels, where 1 represents similar pairs and 0 represents non-similar pairs [141]. Specifically, cross entropy loss is used to compute the loss for image-to-text $L_{i \to t}$ and for text-to-image $L_{t \to i}$, as represented in (24).

$$\mathcal{L}_{i \to t} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} y_{ij} \log p_{ij}^{i \to t}$$
$$\mathcal{L}_{t \to i} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} y_{ij} \log p_{ij}^{t \to i} \tag{24}$$

The final contrastive loss is obtained by computing the average of the two losses, as shown in (25).

$$\mathcal{L}_s = \frac{1}{2} \left(\mathcal{L}_{i \to t} + \mathcal{L}_{t \to i} \right) \tag{25}$$

4.8 Classification Layer

The last layer is the classification layer, which gives the final prediction. It consists of a feedforward network, a classifier, and a softmax function, as illustrated in Figure 4.1 (D). It also uses categorical cross-entropy as the loss function during training, which is combined with the alignment loss from the similarity measurement branch.

The feedforward network consists of one fully connected layer. At first, layer normalization is applied to ensure that all vector values are on the same scale. Following normalization, there is a linear layer. Although the fused vector contains features extracted from MobileNetV3 and TinyBERT, these features are general-purpose and not specific to the task of classifying "real" or fake". Therefore, the linear layer is used to learn what features are most relevant for the task of fake news classification. Next, dropout is applied during training to avoid overfitting, and lastly, a GELU activation function is then applied to introduce non-linearity. This sequence may be repeated multiple times, and in our study, we are experimenting with different repetition counts to determine which configuration remains lightweight while still being effective.

Moreover, the classifier consists of layer normalization, followed by a linear layer, a GELU activation function, and a final linear layer. The output vector from the feedforward network is first normalized and then fed through a linear layer to transform the vector into a new representation. After that, GELU activation function is applied. Lastly, a second linear

layer produces raw scores for each class, called logits. These logits are then passed into a softmax function to convert them into probabilities, as defined in Equation (26),

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \tag{26}$$

where z_i denotes the input logit for class i and C is the total number of classes.

During training, categorical cross-entropy loss function, as defined in Equation (27), where y_i denotes the true label for class i, $\hat{y}_i$ denotes the predicted probability for class i, and C is the total number of classes, is used to measure how close the predicted probabilities are to the true label. This loss is combined with an alignment loss from the similarity measurement branch to model cross-modal inconsistency, as show in (28), where α and β are hyperparameters. The resulting loss value is then propagated back to adjust the model's weights. However, during testing, the softmax function produces the class probabilities, and the class with the highest probability is taken as the final prediction.

$$L(y_i, \hat{y}_i) = -\sum_{i=1}^{C} y_i \log \hat{y}_i \tag{27}$$

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}(y_i, \hat{y}_i) + \beta \mathcal{L}_s, \quad \alpha + \beta = 1 \tag{28}$$

4.9 Parameter-Efficient Cross-Modal Attention Architecture

This section provides a detailed description of the proposed parameter-efficient cross-modal attention architecture. It begins with an overview of the overall design, followed by a description of the modality embeddings and the projection layer. It then introduces the cross-modal attention block, including the learned token pooling mechanism, the cross-modal attention fusion process, and the low-rank weight decomposition strategy. The architecture is illustrated in Figure 4.2.

4.9.1 Design Overview

In the baseline architecture, multimodal fusion is performed using simple operations such as feature concatenation and element-wise summation, which do not explicitly capture cross-modal relationships. On the other hand, cross-modal attention fusion enables the model to capture relationships between different modalities. However, previous studies have shown that this approach is computationally heavy.

Since our goal is to develop a lightweight architecture, using standard cross-modal attention is impractical, because each token in one modality attends to all tokens in the other modality, resulting in quadratic complexity.

To address this challenge, we introduce a parameter-efficient cross-modal attention block that combines low-rank weight decomposition with learned token pooling, as shown in Figure 4.2 (C). This design reduces sequence lengths before cross-attention operations, which decreases the number of parameters and the overall computational cost while preserving cross-modal interactions.

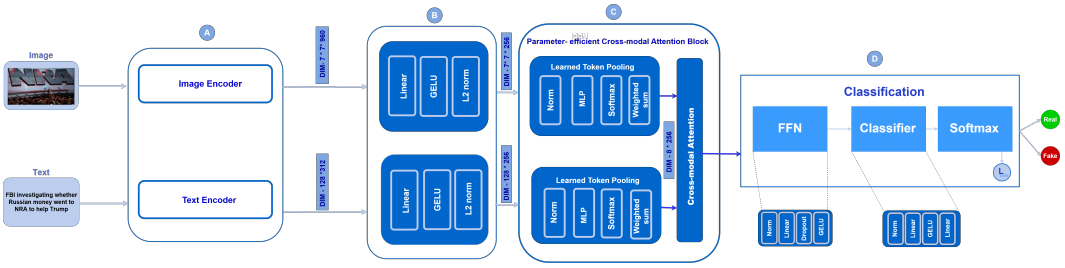


Figure 4.2: Overall architecture of the proposed Parameter-efficient Cross-modal Attention model. **(A)** Image and text encoders. **(B)** Projection head. **(C)** Parameter-efficient Cross-modal Attention Block. **(D)** Classification layer.

4.9.2 Modality Encoders and Token Representations

The proposed architecture employs the same encoders as the baseline architecture, using MobileNetV3 [25] as the image encoder and TinyBERT [15] as the text encoder. Token-level representations are required to perform attention-based fusion. For the image encoder, MobileNetV3, the final classification layer and the global average pooling layer are removed to obtain spatial feature tokens. The resulting feature map has dimensions $7 \times 7 \times 960$, resulting in 49 patches, each represented as a 960-dimensional vector, which we refer to

as visual tokens. For the text encoder, TinyBERT, the output consists of a sequence of contextualized token embeddings. In this project, the maximum sequence length is set to 128, resulting in 128 textual tokens, each with a dimensionality of 312.

4.9.3 Projection Head

The projection head, shown in Figure 4.2 (B), follows the same design as in the baseline architecture described in Section 4.5. The inputs to this layer are the visual tokens of size 49×960 and the textual tokens of size 128×312. Since the two modalities differ in feature dimensionality, both are mapped into a shared latent space of dimension 256 to enable efficient low-rank cross-modal attention. In low-rank attention mechanisms [214], the number of trainable parameters and computational cost scale linearly with the embedding dimension, making dimensionality reduction an important design consideration. Furthermore, prior work employing latent bottleneck attention mechanisms demonstrates that projecting high-dimensional inputs into a moderate latent space, such as 256 or 512, allows attention to remain computationally efficient while maintaining competitive performance compared to larger dimensions [215]. Motivated by these considerations, we adopt a 256-dimensional shared space for cross-modal interaction.

4.9.4 Cross-modal Attention Block

The cross-modal attention block, illustrated in Figure 4.2 (C), enables efficient interaction between visual and textual modalities by combining learned token pooling and low-rank weight decomposition. Token pooling is used to reduce the number of tokens involved in the attention computation, while low-rank decomposition is applied to limit the number of trainable parameters. A detailed description of these components is given in the following sections.

4.9.4.1 Learned Token Pooling

After projection into the shared latent space, the visual and textual modalities are represented as token sequences of lengths 49 and 128, respectively. Applying cross-modal attention directly on these sequences would be computationally expensive. To reduce this cost, we employ a learned token pooling mechanism that compresses each modality into a

small set of representative tokens before cross-modal interaction.

For the visual modality, we adopt the TokenLearner module proposed by Ryoo et al. [216]. Specifically, we use the TokenLearnerModuleV11 variant, which generates a fixed number of learned visual tokens by computing attention maps over the spatial feature representation. Given the projected visual tokens of size $49 \times D$, the TokenLearner module produces $K = 8$ learned tokens of size $8 \times D$ through weighted aggregation. Concretely, TokenLearner learns a set of token-specific attention maps using a lightweight multilayer perceptron, normalizes these maps across spatial locations, and aggregates the input features via weighted summation. This mechanism allows the model to focus on the most informative spatial regions while significantly reducing the number of tokens processed by subsequent attention layers.

Formally, let the projected visual feature map be $X \in \mathbb{R}^{H \times W \times C}$. The i-th learned token is computed as

$$z_i = \rho\left(X \odot \gamma(\alpha_i(X))\right), \tag{29}$$

where $\alpha_i(X)$ generates a spatial attention map, $\gamma(\cdot)$ broadcasts the attention map across channels, $\odot$ denotes element-wise multiplication, and $\rho(\cdot)$ represents spatial global average pooling. As shown in Equation (29), each token is obtained by modulating the input feature map with a learned attention mask and aggregating it spatially.

The final pooled representation is given by equation (30), where $K = 8$ in our architecture.

$$Z = [z_i]_{i=1}^{K} \in \mathbb{R}^{K \times C}, \tag{30}$$

For the textual modality, an attention-based learned pooling mechanism is used to compress the projected textual tokens into a fixed number of $K = 8$ tokens prior to cross-modal fusion. A lightweight feed-forward network computes token-specific attention scores, which are normalized across the sequence using a softmax operation. The pooled textual tokens are obtained by aggregating the tokens through weighted summation [217], resulting in a compact set of representations that are more efficient for cross-modal attention.

Given textual tokens $X \in \mathbb{R}^{N \times D}$, attention weights are computed as shown in equation (31)

$$A = \text{softmax}(f_\theta(X)), \tag{31}$$

Then, the pooled tokens are obtained as shown in equation (32), which compresses the sequence into K learned tokens.

$$Z = AX. \tag{32}$$

4.9.4.2 Cross-modal Attention Fusion

For cross-modal attention, we adopt a lightweight cross-modal attention similar to the one proposed in [214]. This approach introduces only a single learnable projection matrix in order to avoid increasing model complexity. In particular, only the query projection matrix W_Q is learned, while the key and value representations are directly reused from the input feature vectors. As a result, this approach significantly reduces the computational cost while preserving effective cross-modal interactions.

The input to the attention module for each modality consists of 8 tokens, each represented by a 256-dimensional feature vector. Thus, the feature matrix for modality i can be written as:

$$X_i \in \mathbb{R}^{8\times 256}.$$

Taking the text modality as an example, its feature matrix is first passed through a linear transformation to produce the query matrix. However, unlike standard attention, this projection is implemented using a low-rank decomposition strategy, as detailed in Chapter 4.9.4.3.

Then, each text token computes its similarity with all image tokens using the dot-product between the query and key vectors. This produces a pairwise similarity matrix, which is normalized using a row-wise softmax operation to obtain attention weights, as shown in Equation (33):

$$\alpha_{t\to i} = \text{softmax}\left(\tanh\left(\frac{Q_t K_i^\top}{\sqrt{d_k}} \right) \right), \tag{33}$$

where Q_t is the query matrix for text, K_i is the key matrix for the image, and d_k is the feature dimension used for scaling. The row-wise softmax ensures that each text token independently attends to all image tokens.

The attention output for the text modality is obtained by multiplying the attention scores with the value matrix from the image, as shown in Equation (34):

$$A_t = \alpha_{t \to i} V_i, \tag{34}$$

where V_i is the image value matrix.

Subsequently, the resulting attended feature is added to the original text features via a residual connection and normalized, as shown in Equation (35):

$$E_t = \text{Norm}(A_t + X_t), \tag{35}$$

where A_t is the attention output and X_t is the original text feature matrix. This residual with normalization preserves the original text information, stabilizes gradient flow during training, and allows A_t to complement rather than overwrite X_t.

Lastly, in order to obtain the final text modality-aware feature representation enriched with information from the image tokens, a linear transformation followed by a ReLU activation is applied to the normalized attended text features, resulting in the updated text representation, as defined in Equation (36):

$$O_t = \text{ReLU}(\text{Linear}(E_t)). \tag{36}$$

The same process is applied to the image modality, where each image token computes attention weights over all text tokens based on pairwise similarity, enabling the model to capture cross-modal relationships between the two modalities.

After obtaining the modality-aware representations for both text and image, the final cross-modal representation is constructed by fusing O_t and O_i, as shown in Equation (37). Specifically, the two representations are combined using element-wise addition to produce the final fused matrix:

$$F = O_t + O_i, \tag{37}$$

where F denotes the final fused cross-modal representation. This cross-modal attention fusion strategy, as illustrated in Figure 4.3, preserves complementary information from both

modalities while maintaining computational efficiency.

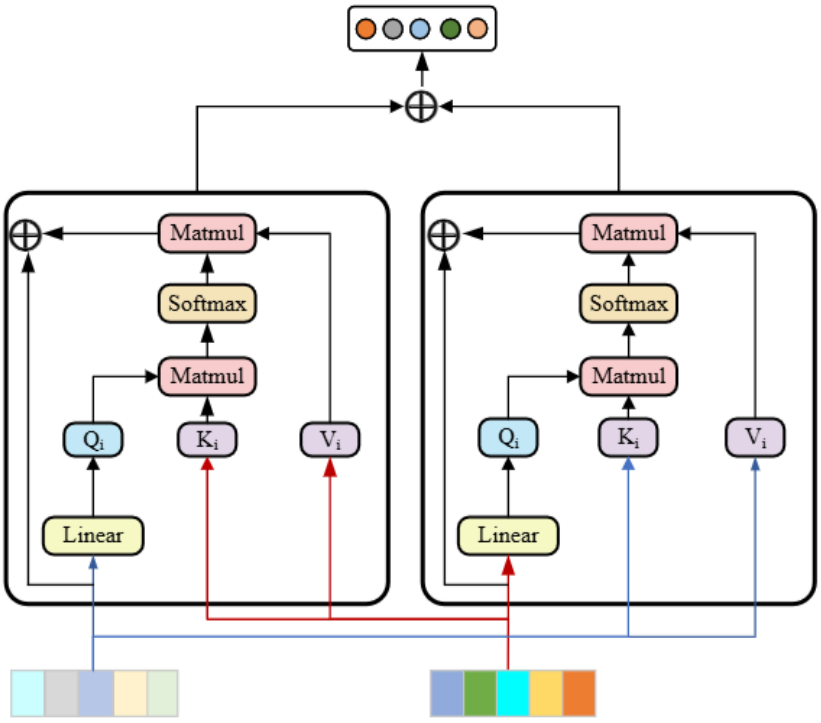


Figure 4.3: Cross-modal Attention Fusion.

4.9.4.3 Low-rank Weight Decomposition

Low-rank weight decomposition is a technique used to reduce the number of trainable parameters and compress model complexity by factorizing a weight matrix into the product of two or more smaller matrices [218].

As mentioned earlier, we follow a similar approach to that proposed in [214], where only the query matrix W_Q is learned. In this approach, the query weight matrix is computed using low-rank decomposition, while the key and value matrices are directly reused from the input features. Specifically, for each modality $i \in \{t, i\}$, the query matrix is expressed as a sum of low-rank factors, as shown in Equation (38):

$$W_Q^{(i)} = \sum_{j=1}^{r} \omega_j^{(i)}, \quad \omega_j^{(i)} \in \mathbb{R}^{d_n \times d_q}, \tag{38}$$

where r denotes the rank of the decomposition, $\omega_j^{(i)}$ are the low-rank factor matrices specific to modality i, d_n is the input feature dimension, and d_q is the query dimension.

Using this decomposition, the query matrix for modality i is computed as defined in Equation (39):

$$Q_i = \left(\sum_{j=1}^{r} \omega_j^{(i)} \right) X_i + b_{Q_i}, \tag{39}$$

where X_i is the input feature matrix for modality i (either X_t for text or X_i for image), and b_{Q_i} is the bias term. This decomposed query matrix Q_i is then used in the attention computation as described in Equation (33), producing attention weights $\alpha_{t \to i}$ for text-to-image or $\alpha_{i \to t}$ for image-to-text interactions.

By applying low-rank decomposition to both text and image query matrices, the attention mechanism becomes more computationally efficient while still capturing cross-modal relationships between the two modalities.

4.10 Summary

In this chapter, a detailed description of both architectures was provided. Both use TinyBERT as the text encoder and MobileNetV3 as the image encoder. In the baseline architecture, the resulting embeddings are passed through a projection head to map them into a common dimensional space and then fused using simple feature-level operations before being fed to the classification layer. In the parameter-efficient cross-modal attention architecture, the encoders produce token sequences instead of embeddings, which are also projected into the same dimensional space using a projection head, as in the baseline. These tokens are then used as input to the parameter-efficient cross-modal attention block, which applies learned token pooling and cross-modal attention with low-rank weight decomposition. The resulting modality-aware representations are fused and passed to the classification layer to produce the final prediction.

Chapter 5: Experimental Design

This chapter presents the experimental design of our project. First, the overall experimental procedure is mentioned, along with the datasets we will employ in our project and the preprocessing steps taken to prepare both the visual and textual modalities. Next, we describe the preliminary experiments conducted to select the appropriate image and text encoders, which include the experimental protocols and the findings that led to our choices. Lastly, we explain the training process, the steps that will be taken to evaluate our framework, the ablation study conducted to analyze the impact of the architectural components, and the implementation environment.

5.1 Experimental Procedure

The experimental procedure followed in this project to solve the problem of multimodal fake news detection can be divided into several phases. First, preprocessing was done to both the textual and visual modalities in the datasets accordingly. The preprocessing steps include normalization and resizing for images, while tokenization, truncation, and padding for text. After preprocessing, the datasets are partitioned into three subsets: a training subset for training the model, a validation subset for monitoring the model's performance, and a testing subset for final evaluation of the model. The next phase in the pipeline is constructing the architecture. First, to select the appropriate encoders, a preliminary experiment was conducted, and the results show that TinyBERT and MobileNetV3 achieved the best performance among the tested encoders. Therefore, the architecture proposed in this project adopts TinyBERT as the text encoder and MobileNetV3 as the image encoder. The output embeddings of these encoders then go through identical projection heads to map them into the same dimensional space. These compatible embeddings are then fused using several different fusion strategies, such as concatenation and element-wise maximum. Lastly, the unified representation goes through a classification layer to obtain the final output. After constructing the architecture, the next phase is to tune the hyperparameters using the validation set to select the best model configuration. Additionally, the model is trained end-to-end using the training dataset. The final phase is model evaluation, where the multimodal framework is evaluated on the validation set and tested on the test set to

examine its overall performance. Additional details of datasets, data preprocessing, and the proposed architecture are provided in Chapters 4 and 5.

Based on the previously proposed research questions and pipeline this project follows, we formulate the following hypotheses. First, we hypothesize that a lightweight architecture can effectively be utilized in the field of multimodal fake news detection while achieving competitive performance. Moreover, we expect that by employing different fusion strategies, such as concatenation and element-wise maximum, the performance of multimodal fake news detection will differ significantly.

5.2 Datasets

This section provides a detailed description of the datasets that will be employed in this project. These datasets are: Fakeddit [23] and Twitter MediaEval [24], which are two common multimodal datasets in the field of fake news detection.

5.2.1 Fakeddit dataset

One of the datasets used in our study is the Fakeddit dataset [23], which is a multimodal dataset that contains text, images, comments, and metadata. It is sourced from Reddit, a social media platform, where users can post submissions on various subreddits and each subreddit has its own theme. In the context of this dataset, a submission refers to a Reddit post that contains a title, the corresponding image, if available, and metadata such as the subreddit, score, author, and comments. Importantly, the text consists of only the image's caption, and it does not include any accompanying article. Moreover, all submissions were collected using the pushshift.io API.

The dataset contains 1,063,106 samples collected from 22 subreddits. Among these, 682,661 are multimodal samples containing text and image pairs, 413,753 are fake samples, and 268,908 are real samples, as shown in Figure 5.1. Moreover, the released dataset indicates a split of approximately 82.6% training, 8.7% validation, and 8.7% testing, as illustrated in Figure 5.2.

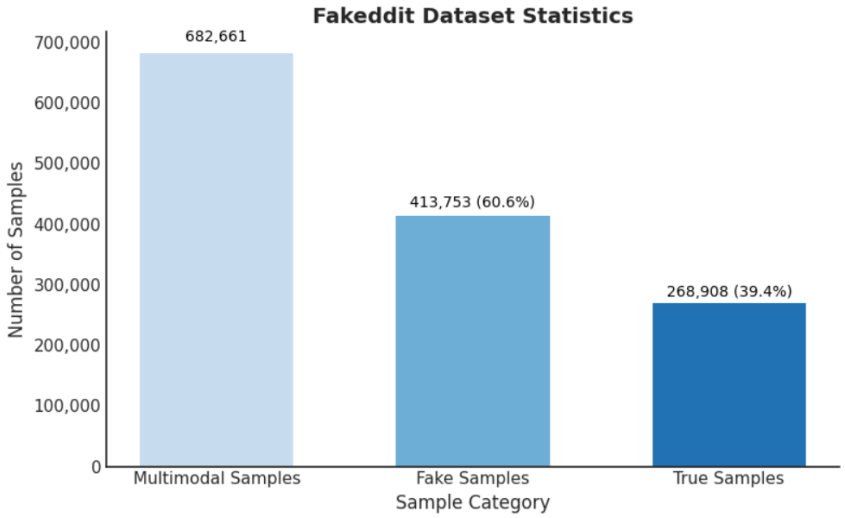


Figure 5.1: Fakeddit Dataset Class Distribution.

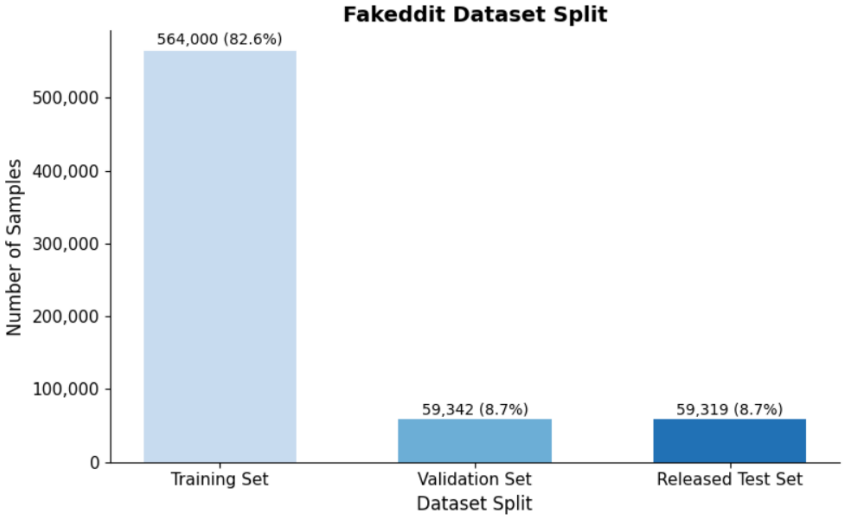


Figure 5.2: Fakeddit Dataset Split.

In addition, each sample in the dataset has three distinct label columns corresponding to the 2-way classification, 3-way classification, and 6-way classification, allowing the 2-way label to be utilized directly without any preprocessing. For the 2-way classification, a sample is either true or fake. For the 3-way classification, a sample is either completely true, fake and contains text that is true, or fake with false text. Lastly, for the 6-way classification, a sample is labeled as either true, satire/parody, misleading content, imposter

content, false connection, or manipulated content. True label indicates that both the text and image are accurate. Satire/parody refers to posts that are designed for humor or satire and not meant to convey factual information. Misleading content describes posts that are intentionally manipulated to mislead the audience, while imposter content indicates that the post contains bot-generated content. False connection applies to post where the image doesn't support the corresponding text, and lastly, manipulated content indicates that the image has been altered or edited. When utilizing the 2-way classification, a sample is labeled true only when the image and the corresponding text are true and related, and all other cases are labeled as false. In Table 5.1, several samples from the dataset using the 2-way classification are presented.

Although the dataset provides metadata, comments, and multiple labeling schemes, we will only focus on using text, images, and the 2-way classification.

ID	Multimodal Sample (Image & Text)	Label
1	 Donald Trump threatens 'fury' against N Korea – BBC News	Real
2	 Gross! 7 Celebrities Who Are 45 Or Older	Fake
3	 Dogs cannot get 'autism', British Veterinary Association warns after 'anti-vaxx' movement spread to pets	Real
4	 Wounded Italian World War 2 soldier, left almost unrecognizable due to the numerous severe injuries received in combat. (1941)	Fake

Table 5.1: Image–Text Samples from Fakeddit Dataset (Binary Classification)

5.2.2 Twitter MediaEval dataset

In addition, we will employ the Twitter MediaEval dataset (2016) [24], which is a multimodal dataset that consists of social media posts collected from Twitter. Each post includes a short text, a corresponding image or video, and metadata such as username and timestamp, which refers to the time the tweet was posted. Furthermore, the posts were collected using the Twitter API and visual near-duplicate search strategy, which involves first collecting posts using keywords related to 17 specific events, then they used the manually verified sets of fake and real images that are related to the events as visual queries to find posts containing images that are either exact matches or near-duplicates of the verified images.

The dataset is split into two sets, the development set and the test set. Each set contains real and fake posts related to specific cases that are related to 17 major events such as Hurricane Sandy, Boston Marathon bombing, Sochi Olympics, and others. For the development set, which is equivalent to a training set, it includes 193 real cases with 6225 associated real posts posted by 5895 unique users and 220 fake cases with 9596 fake posts posted by 9216 unique users. The total number of posts in the development set is 15821. As for the test set, it is composed of 991 real posts and 1186 fake posts, resulting in a total of 2177 posts. The distribution of the dataset is illustrated in Figure 5.3, and the dataset split is presented in Figure 5.4. Since the dataset contains posts in multiple languages and includes posts with videos, we filtered out the non-English posts, removed the posts containing videos, and re-split the remaining posts into development and test sets, resulting in 10026 posts in the development set and 2501 posts in the test set, which correspond approximately to 80% and 20% of the total 12,527 English posts with only images.

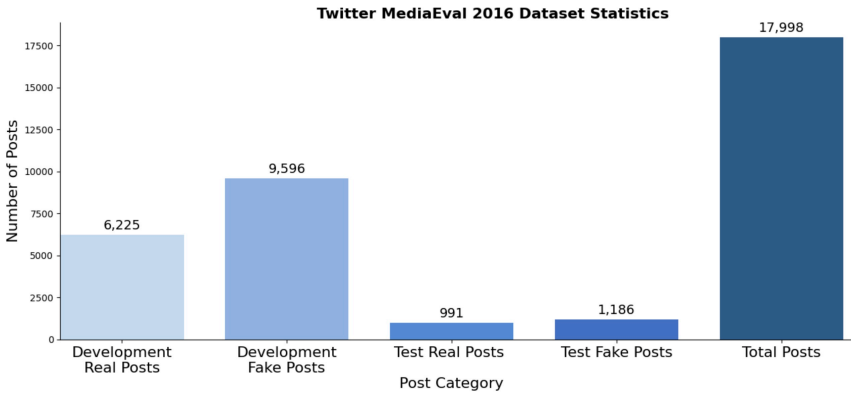


Figure 5.3: Twitter MediaEval Class Distribution.

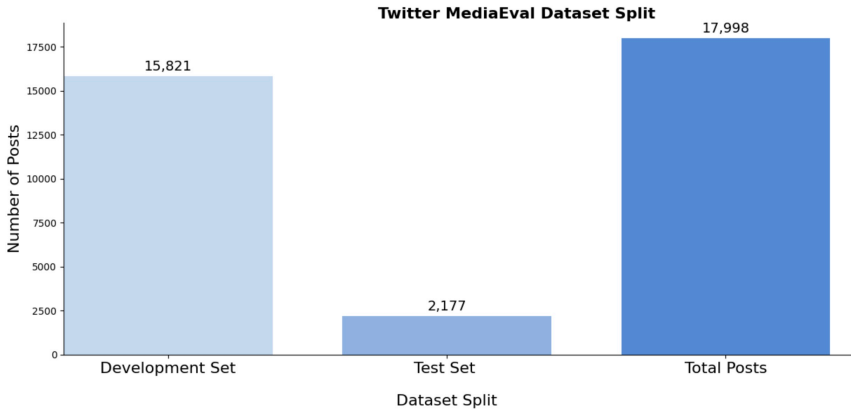


Figure 5.4: Twitter MediaEval Dataset Split.

Additionally, each post in the dataset has two labels, either real or fake. Fake posts consist of an image or video that does not relate to the event it claims to represent in the text. Moreover, real posts include images or videos that accurately relate to the event described in the text. Table 5.2 shows a sample of the dataset.

ID	Multimodal Sample (Image & Text)	Label
1	 BREAKING: FRANCE24 reports at least 60 hostages at #Paris concert hall	Real
2	 North Korea's Nuclear Missile Threat: Very Bad News	Fake
3	 Thanks for the Turk Schools in Nepal for rescuing the victims and helping them in their relief.	Real
4	 @UberrFootbaII: In Germany people take to the streets to shout "The Germany is with the France"	Fake

Table 5.2: Image–Text Samples from Twitter MediaEval Dataset (Binary Classification)

5.3 Data Preprocessing

Data preprocessing is a crucial stage that prepares raw data for computational analysis, directly affecting the results of the models. Whether dealing with text or images, both modalities require several preprocessing techniques to enhance their quality and effectively be used for the task of fake news classification. To ensure a fair evaluation, preprocessing must be applied on the entire dataset.

5.3.1 Image Data Preprocessing

Image preprocessing involves several steps to improve the overall representation quality and reliability of visual data. Below, the fundamental image preprocessing procedures employed in this project are mentioned.

- **Image resizing:** Resizing refers to scaling an image into specific dimensions so they are compatible with the image encoder. Images are resized to 224×224.
- **RGB conversion:** RGB conversion refers to converting every input image into three color channels (Red, Green, and Blue) to ensure consistency across the dataset.
- **Tensor Conversion:** Images are converted into multidimensional numerical arrays, known as tensors, so they could be used as input to the pretrained image encoder [219].
- **Image Normalization:** All input images were normalized using the ImageNet dataset mean and standard deviation to match the input expected by the pretrained image encoder.

As for training data, additional data preprocessing steps, such as random cropping and color jittering, are applied. These allow the image encoder to generalize better and avoid positional and lighting bias, respectively [220].

5.3.2 Text Data Preprocessing

Text preprocessing is done in several steps to clean and ensure that the textual data is formatted correctly for the model to process it. The main text preprocessing steps are:

- **Lowercasing:** All the text is converted to lowercase to ensure consistency in the text and to reduce the number of unique words.
- **Data Cleaning:** Irrelevant elements such as URLs, emojis, special characters, and duplicates are removed from the text to reduce noise and maintain meaningful information.
- **Tokenization:** Tokenization refers to dividing text into smaller meaningful units

using the WordPiece algorithm [221], which helps the encoder to handle new words by splitting them into subword units it already recognizes. Traditional NLP preprocessing techniques like stemming and lemmatization are not required in BERT-based encoders such as TinyBERT, since WordPiece tokenization already handles different variations [222].

- **Special Token Addition:** Special tokens such as [CLS] and [SEP] are added to the beginning and end of each text sequence to indicate its boundaries.
- **Conversion to IDs:** Tokens are converted to numerical identifiers based on the encoder's vocabulary, so they can be represented numerically.
- **Padding and Truncation:** Each sequence is adjusted to a fixed length of 128 by truncating longer inputs and adding padding to shorter ones with [PAD] tokens.
- **Attention Mask Generation:** Attention masks are generated to identify actual tokens and ignore padding tokens during encoder computation.
- **Tensor Conversion:** The processed text is converted into tensors, which are multi-dimensional numerical arrays, so that they could be used as input to the encoder.

5.4 Preliminary Experimentation for Encoder Selection

In order to select the appropriate encoder for each modality, a comparison was carried out independently for the text and image models. The models selected for comparison were based on their potential of being computationally efficient and lightweight, while maintaining high accuracy. Below, the experimentation details for image and text models are described independently.

5.4.1 Image Encoder Selection

The models selected for comparison are the image transformer model MobileVit-V2 [71] from the Hugging Face Transformer library [223], and the CNN-based models GhostNetV2 [14], and MobileNetV3-Large [25], both obtained from the TIMM PyTorch Image library

[224]. The experiment was done on the first 10% of the Fakeddit dataset [23] which included approximately 37,600 non-corrupted images.

5.4.1.1 Experimental Protocols

Starting with image preprocessing, each model followed the default preprocessing pipeline provided by its library. The images that are used as input for the MobileVit-V2 model were processed using the Hugging Face AutoImageProcessor, which resizes images to 256×256 pixels, converts them to RGB format, and rescales them to normalize the pixel values to $[0, 1]$. On the other hand, the images that are fed into the CNN-based models were resized to 224×224 pixels, converted into a tensor, and normalized using the ImageNet mean and standard deviation. Table 5.3 shows the training configuration used on all the image-based models to conduct a fair and consistent evaluation.

Table 5.3: Training configuration used for all image-based models

Hyperparameter	**Value**
Epochs	5
Batch Size	32
Learning Rate	1×10^{-4}
Loss Function	Cross-Entropy Loss

The models were then trained on the sample dataset, with 80% of the data used for training and 20% for validation to evaluate the models. Table 5.4 presents the training time required to complete 5 epochs and the corresponding accuracy achieved by each model. The experiments were implemented in Google Colab using a NVIDIA T4 GPU.

Table 5.4: Comparison of Transformer-based and CNN-based models

Category	**Name**	**Params (M)**	**FLOPs (G)**	**Acc (%)**	**Layers**	**Training Time (min)**
CNN-based	GhostNetV2	4.88	0.362	75.69	524	29
CNN-based	MobileNetV3-Large	5.48	0.22	74.76	296	26
Transfomer-based	MobileVit-V2	4.39	3.70	75.47	286	40

5.4.1.2 Findings

From the comparison shown in Table 5.4, although MobileViT-V2 achieved competitive accuracy, it required a much higher computational cost due to its large FLOPs value, which also resulted in the longest training time among the models. Since our focus is to build an efficient and lightweight architecture, MobileViT-V2 was not considered a suitable option for our model. This left the comparison mainly between GhostNetV2 and MobileNetV3. While GhostNetV2 achieved the highest accuracy, the improvement compared to MobileNetV3 was less than 1%, which is not a significant difference. Moreover, GhostNetV2 required slightly more computation and about 3 minutes longer training time on the selected dataset sample. In contrast, MobileNetV3 reached a similar accuracy while having lower FLOPs and faster training. Therefore, considering both performance and efficiency, MobileNetV3 offers a better trade-off between accuracy, model size, and computational cost, making it the most practical choice to serve as the image encoder in our model.

5.4.2 Text Encoder Selection

For the text model selection, several Transformer models were compared, including ALBERT-base [225], TinyBERT [15], DistilBERT [226], and MiniLM [16], from the Hugging Face Transformers library [223], along with DeBERTa-v3 [66] with Low-Rank Adaptation (LoRA) [61]. The experiment was performed on a sample from the Fakeddit dataset [23] which consisted of 470,754 text samples.

5.4.2.1 Experimental Protocols

For text preprocessing, text containing two or fewer words was removed. Then, the standard tokenization pipeline provided by the Hugging Face library [223] was used for all the models. The text inputs were tokenized using each model's AutoTokenizer with a maximum sequence length of 128 tokens across all models, including ALBERT, TinyBERT, DistilBERT, MiniLM, and DeBERTa-v3 with standard LoRA configuration, and padding was applied as needed along with truncation. After tokenization, the input was converted to input IDs, and attention masks were created to mark which part of the text is real input

and which is padding. Then, the data was formatted as PyTorch tensors, which is the data format required by the model for training.

Table 5.5: Training configuration used for all text-based models

Hyperparameter	**Value**
Epochs	3
Batch Size	16
Learning Rate	2×10^{-5}
Loss Function	Cross-Entropy Loss
Max Sequence Length	128

After preprocessing, the data was split into 80% training, 10% validation, and 10% testing for model evaluation. The training time required for 3 epochs and the achieved accuracy are represented in Table 5.6. The experiments were implemented in Google Colab using a NVIDIA T4 GPU.

Table 5.6: Comparison of Transformer-based text models Note: The symbol "–" indicates that the FLOPs value for MiniLM is unavailable and not officially reported in the original paper.

Name	**Params (M)**	**FLOPs (G)**	**Acc (%)**	**Layers**	**Training Time (min)**
ALBERT	12	21.8	85.52	12	186
TinyBERT	14.5	1.2	86.39	4	35
MiniLM	22	–	87.02	6	43
DistilBERT	66	22	87.13	6	92
DeBERTa-v3 with base LoRA	184	50	85.62	12	161

5.4.2.2 Findings

Based on the results shown in Table 5.6, TinyBERT achieved the most optimal trade-off between accuracy, model size, and computational efficiency. Although MiniLM showed a slightly better accuracy of 87.02% than TinyBERT 86.39%, it had a larger number of parameters (22M) compared to TinyBERT (14.5M), making TinyBERT more suitable for lightweight deployment. Moreover, MiniLM required more training time and was slower,

whereas TinyBERT achieved this accuracy with fewer parameters and lower computational complexity. In comparison, the larger models DistilBERT (66M) and DeBERTa-v3 (184M) achieved competitive results but at a much higher computational cost. ALBERT, while being the smallest model in terms of parameters, performed worse in accuracy 85.52% and took substantially more time to train (186 minutes). Therefore, TinyBERT was selected as the text encoder for our multimodal model because it provides competitive accuracy with smaller model size, faster training, and lower computational cost aligning with the goal of building a lightweight and efficient architecture.

5.5 Model Selection

After constructing the architecture proposed in Chapter 4, we will perform end-to-end training using the training dataset. In order to obtain the optimal overall model with the best results, some hyperparameters are tuned on the validation set of the data, while others are kept fixed. The tunable hyperparameters in our model include the batch size, while a maximum epoch value, such as 100, 200, or 250, will be selected with early stopping. For the classification layer, we will experiment with the number of blocks in the feedforward network, each containing layer normalization, a linear layer, dropout, and GELU activation. We will also vary the dropout rate in both the projection head and the classification layer. Moreover, the model will be trained using the AdamW optimizer [227] while testing different learning rates. These hyperparameters are tuned using Optuna [228], which is a hyperparameter search software that finds the suitable hyperparameter configuration. Furthermore, several fusion strategies, mentioned in Chapter 4, will be tested to determine which approach captures the cross-modal interactions between the text and image modalities efficiently. Table 5.7 shows the range of hyperparameter values to be experimented with.

Table 5.7: Hyperparameter Tuning for Experimentation

Hyperparameter	Range of Values
Batch Size	32, 64, 128
No. of Blocks in the Feedforward Network of the Classification Layer	1 block, 2 blocks
Dropout Rate	0.1 , 0.2 ,0.4
Learning Rate	1e-2, 1e-3, 1e-4
Fusion Strategies	Concatenation, Element-Wise Max, Siamese Style, Hadamard Product with Element-Wise Sum
Loss Weights (α, β)	(0.8,0.2), (0.7,0.3)

As for the fixed hyperparameters, Table 5.8 shows the hyperparameters and their fixed values.

Table 5.8: Fixed training configurations

Hyperparameter	Value
Projection Head Dimension	512
Projection Head Layer	1 Layer
Projection Head Activation	GELU
Classification Layer Activation	GELU
TinyBERT Token Length	128
Optimizer Type	Adam Optimizer
Loss Function	Cross-Entropy Loss

5.6 Model Evaluation

This section presents the evaluation steps used to assess the performance of our proposed framework in fake news detection and to ensure that the framework is evaluated fairly. Firstly, we follow the available training, validation, and test splits provided by the Fakeddit dataset to ensure a reliable evaluation of performance. Additionally, we will employ the Twitter MediaEval dataset and follow an 80% training, 10% validation, and 10% testing split. Also, we will perform a cross-dataset evaluation using the Twitter MediaEval dataset, to assess our framework's ability to generalize across different datasets. Moreover, we will use several evaluation metrics, including accuracy, precision, recall, F1-score, FLOPs, and number of parameters to report the performance of our framework. The detailed explanations of these metrics are provided in Chapter 2.

5.7 Ablation Study Design

As part of the experimental design we perform an ablation study to determine how each component affects the performance of the proposed multimodal model. Even though the performance metrics summarize the effectiveness of the entire architecture, the ablation experiments allow us to isolate and evaluate the role of specific components.

In this study, we remove selected components from the architecture to analyze their impact on performance. We design five ablation experiments, each focusing on a specific modality or architectural element.

5.7.1 Image-Only Model

Within the ablation design of this study, we remove the textual branch entirely and rely only on the image modality. The input image is processed using MobileNetV3, followed by the projection head. The resulting image features are directly fed into the classification layer without any fusion. This setup enables us to assess the independent contribution of visual features to the overall system performance.

5.7.2 Ablation Using Text Only

In this ablation setting, we remove the image branch and rely on the textual modality only. Specifically, the image encoder, MobileNetV3 [25] is excluded, and only the text encoder, TinyBERT [15], is used to extract the meaningful textual features from the input news text. The textual features are then passed through the projection head and directly fed into the classification layer, without any multimodal fusion. This setup allows us to evaluate the effectiveness of TinyBERT in detecting fake news and to examine whether a multimodal approach, which combines both text and images, performs better than a unimodal approach.

5.7.3 Impact of Fully Connected Layers With Shared Weights

In this part, we investigate the impact of introducing an additional fully connected layer after the projection heads of both the image and the text modalities. In this configuration, the fully connected layers share the same weights in order to obtain a more similar representation, as done in [213]. The resulting output vectors are then forwarded to the similarity measurement branch, where a similarity score is computed and used to derive the

alignment loss. This ablation allows us to evaluate whether enforcing fully connected layers with shared weights after the projection heads improves cross-modal alignment.

5.7.4 Similarity Measurement branch

In this ablation experiment, we remove the similarity measurement branch from the proposed architecture. In this setup, no similarity score is computed between the projected image and text embeddings, and the similarity loss is not used during training. The fused image and text embeddings are directly passed to the classification layer. This experiment allows us to evaluate the impact of the similarity measurement branch on the overall model performance.

5.7.5 Projection Head Replacement (PCA)

As part of the ablation study, we replace the learnable projection head with Principal Component Analysis (PCA). The output embedding dimension is kept the same as in the original architecture to ensure a fair comparison and isolate the effect of learning a task-specific projection. Unlike the projection head, PCA applies a fixed linear transformation without trainable parameters. This experiment evaluates whether the performance gains arise from the learned projection or whether a non-trainable linear mapping is sufficient.

5.8 Implementation Environment

We will implement our experiments using Google Colab, which makes it easy to collaborate and share work through Google Drive. Colab is a cloud-based Jupyter notebook environment that requires no local setup, runs entirely online, and provides free GPU runtimes for efficient model training. Python will be our main programming language, taking advantage of its wide range of libraries to develop, run, and evaluate the different parts of the project. As for the hardware used, it is presented in Table 5.9.

Table 5.9: List of Devices Used in the Project

Device	**Operating System**	**Processor**	**Memory**
Windows Laptop (2023)	Windows 11	Intel Core i7-1360P (13th Gen, 2.20 GHz)	16 GB RAM and 954 GB SSD
MacBook (2020)	macOS Sequoia 15.3.1	Apple M1 (8-core CPU, 8-core GPU, 16-core Neural Engine)	8 GB unified memory and 512 GB SSD

5.9 Summary

This chapter describes the experimental design of our project. The datasets we will employ to evaluate and test our framework are Fakeddit and Twitter MediaEval. Additionally, the preprocessing steps of both the image and text modalities are mentioned, along with the preliminary experimentation conducted to find the suitable text and image encoders for our architecture. Lastly, the model selection, evaluation, ablation study design, and implementation environment are described.

Chapter 6: Conclusion

In conclusion, the spread of fake news in recent years have posed a serious threat to individuals and society by spreading news that can influence opinions and cause real-world harm. Also, fake news often combines more than one modality which increases its complexity and potential to mislead the audience. Because of the vast amount of information shared online, and the fact that fake news often appears across multiple modalities, such as text and images, manual detection is no longer practical. This highlights the need for automated systems that are capable of accurately identifying fake news across different modalities. Many studies have relied on a unimodal approach which use only a single type of modality, such as text or images only. However, this approach is often ineffective for detecting fake news, as they cannot capture the complexity of content that spans multiple modalities. Furthermore, while many multimodal approaches have been proposed to improve fake news detection and achieve higher accuracy, they are often computationally expensive and rarely examine different fusion strategies. This research addresses these challenges by proposing a lightweight multimodal architecture that employs MobileNetV3 as the image encoder and TinyBERT as the text encoder to ensure low computational cost. The extracted features from the encoders are then mapped into the same dimensional space using a projection head and are fused using different fusion strategies to identify the most effective approach for combining the textual and visual features. Finally, the fused vector is passed through a classification layer to produce the final prediction. This approach aims to achieve an optimal balance between performance and computational efficiency.

References

[1] S. Munusamy, K. Syasyila, A. H. Shaari, M. A. Pitchan, M. R. Kamaluddin, and R. Jatnika, "Psychological factors contributing to the creation and dissemination of fake news among social media users: a systematic review," *BMC Psychology*, vol. 12, p. 673, Nov. 2024.

[2] E. Denniss and R. Lindberg, "Social media and the spread of misinformation: infectious and a threat to public health," *Health Promotion International*, vol. 40, no. 2, p. daaf023, Mar. 2025. [Online]. Available: https://doi.org/10.1093/heapro/daaf023

[3] R. Bhattacharjee, M. A. Sufian, M. Talha, M. Ahmed, F. Tasnim, S. Sara, and K. M. Islam, "Title-based fake news detection using lstm," in *Proc. Int. Conf. Computer and Information Technology (ICCIT)*, Dec. 2024, pp. 173–178.

[4] P. Sharma and R. Sahu, "Fake news detection using deep learning based approach," in *Proc. Int. Conf. Circuit Power Comput. Technol. (ICCPCT)*, 2023, pp. 651–656.

[5] D. R. Salem, A. A. Abdullah, A. A. AlHabshy, and K. A. ElDahshan, "Detecting fake news images using a hybrid cnn-lstm architecture," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 7, 2025. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2025.0160719

[6] L. S. Alqurashi, S. AlMuraytib, R. Qarout, and N. Zamzami, "Fake image detection in fake news using convolutional neural network (cnn)," in *Proc. Int. Conf. Innovation in Artificial Intelligence and Internet of Things (AIIT)*, May 2025, pp. 1–8.

[7] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," https://arxiv.org/abs/1805.04953, 2018, accessed: Sep. 29, 2025.

[8] I. Segura-Bedmar and S. Alonso-Bartolome, "Multimodal fake news detection," *Information*, vol. 13, no. 6, 2022. [Online]. Available: https://www.mdpi.com/2078-2489/13/6/284

[9] S. K. Uppada, P. Patel, and S. Bhatnagar, "An image and text-based multimodal model for detecting fake news in osns," *J. Intell. Inf. Syst.*, vol. 61, no. 2, pp. 367–393, Sep. 2023.

[10] Y. Guo, B. Li, K. Zhen, J. Liu, G. Li, Q. Wang, and Y.-J. Liu, "Consistency-heterogeneity balanced fake news detection via cross-modal matching," *IEEE Transactions on Artificial Intelligence*, vol. 6, no. 7, pp. 1787–1796, Jul. 2025.

[11] D. A. Mura, M. Usai, A. Loddo, M. Sanguinetti, L. Zedda, C. Di Ruberto, and M. Atzori, "Is it fake or not? a comprehensive approach for multimodal fake news detection," *Online Social Networks and Media*, vol. 47, p. 100314, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468696425000151

[12] H. Wang, Y. Yang, S. Han, Z. He, and W. Yang, "A multimodal fake news detection model based on cross-image semantic fusion," in *Proc. IEEE Int. Conf. on Mobile Ad Hoc and Sensor Networks (MSN)*, Dec. 2024, pp. 738–745.

[13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

[14] Y. Tang, K. Han, J. Guo, C. Xu, C. Xu, and Y. Wang, "Ghostnetv2: Enhance cheap operation with long-range attention," 2022. [Online]. Available: https://arxiv.org/abs/2211.12905

[15] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," 2020. [Online]. Available: https://arxiv.org/abs/1909.10351

[16] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *arXiv preprint arXiv:2002.10957*, 2020.

[17] R. Singh, H. Suyal, and M. Digra, "An approach to detect fake news based on machine learning," in *Proc. 2024 8th Int. Conf. Parallel, Distributed and Grid Computing (PDGC)*, 2024, pp. 239–243.

[18] S. Khare, P. Singh, P. Kumar, and R. K. Shrivastava, "Machine learning approach to detect fake news," in *Proc. 2023 IEEE Int. Conf. Adv. Networking and Telecommunications Systems (ANTS)*, 2023, pp. 1–5.

[19] R. Seetharaman, M. Tharun, S. S. S. Mole, and K. Anandan, "Analysis of fake news detection using machine learning technique," *Materials Today: Proceedings*, vol. 51, no. 8, pp. 2218–2223, 2022.

[20] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake news detection using machine learning ensemble methods," *Complexity*, vol. 2020, pp. 1–11, Oct. 2020.

[21] J. Alghamdi, L. Yuqing, and S. Luo, "A comparative study of machine learning and deep learning techniques for fake news detection," *Information*, vol. 13, no. 12, p. 576, Dec. 2022.

[22] R. Mohawesh, H. Al-Bahadili, A. Qaralleh, A. Al-Dhaqm, and A. A. Al-Zoubi, "Truth be told: A multimodal ensemble approach for enhanced fake news detection in textual and visual media," *Journal of Big Data*, vol. 12, no. 1, Aug. 2025.

[23] K. Nakamura, S. Levy, and W. Y. Wang, "r/fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection," *CoRR*, vol. abs/1911.03854, 2019. [Online]. Available: http://arxiv.org/abs/1911.03854

[24] MediaEval Benchmark, "Mediaeval 2016: Twitter dataset, multimedia evaluation benchmark," Multimedia Evaluation Benchmark, 2016, [Accessed: Sept. 20, 2025]. [Online]. Available: http://www.multimediaeval.org/mediaeval2016/

[25] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," 2019. [Online]. Available: https://arxiv.org/abs/1905.02244

[26] J. Cobbe, "Algorithmic censorship by social platforms: Power and political control," *Philosophy & Technology*, vol. 34, no. 4, pp. 1–25, 2021.

[27] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge University Press, 2008.

[28] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Pearson, 2008.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1097–1105.

[30] T. Kumar, R. Brennan, A. Mileo, and M. Bendechache, "Image data augmentation approaches: A comprehensive survey and future directions," *IEEE Access*, vol. 12, pp. 187 536–187 571, 2024.

[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[32] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, 2011.

[33] P.-O. Siebers, Y. Li, and U. Aickelin, "A survey of text representation methods and their genealogy," *arXiv preprint arXiv:2211.14591*, 2022. [Online]. Available: https://arxiv.org/abs/2211.14591

[34] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, p. 613–620, Nov. 1975. [Online]. Available: https://doi.org/10.1145/361219.361220

[35] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," vol. 14, 01 2014, pp. 1532–1543.

[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013. [Online]. Available: https://arxiv.org/abs/1301.3781

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, [Online]. Available: https://arxiv.org/abs/1810.04805.

[38] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[39] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, Nov. 1958.

[40] I. Goodfellow, Y. Bengio, and A. Courville, *Deep feedforward networks*. Cambridge, MA, USA: MIT Press, 2016, ch. 6, pp. 163–220.

[41] H. Wang and B. Raj, "A survey: Time travel in deep learning space: An introduction to deep learning models and how deep learning models evolved from the initial ideas," *arXiv preprint arXiv:1510.04781*, 2015. [Online]. Available: https://doi.org/10.48550/arXiv.1510.04781

[42] I. Goodfellow, Y. Bengio, and A. Courville, *Convolutional networks*. Cambridge, MA, USA: MIT Press, 2016, ch. 9, pp. 326–366.

[43] M. Taye, "Theoretical understanding of convolutional neural network: concepts, architectures, applications, future directions," *Computation*, vol. 11, no. 3, p. 52, Mar. 2023.

[44] V. Phung and E. Rhee, "A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets," *Applied Sciences*, vol. 9, no. 21, p. 4500, 2019.

[45] S. H. Khan and R. Iqbal, "A comprehensive survey on architectural advances in deep cnns: challenges, applications, and emerging research directions," *arXiv preprint arXiv:2503.16546*, 2025. [Online]. Available: https://arxiv.org/abs/2503.16546

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[47] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 2015, pp. 448–456.

[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015, [Online]. Available: https://arxiv.org/abs/1409.1556. [Online]. Available: https://arxiv.org/abs/1409.1556

[49] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[50] S. A. Ahmad, M. H. Al-Kadi, and R. F. R. Saeed, "Breast cancer screening using convolutional neural network and follow-up digital mammography," in *Proceedings of the International Conference on Computer and Applications (ICCA)*, Beirut, Lebanon, 2018, pp. 272–277.

[51] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Machine Learning (ICML)*, Long Beach, CA, USA, 2019, pp. 6105–6114.

[52] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2019. [Online]. Available: https://arxiv.org/abs/1801.04381

[53] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," 2019. [Online]. Available: https://arxiv.org/abs/1709.01507

[54] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," 2019. [Online]. Available: https://arxiv.org/abs/1807.11626

[55] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," 2020. [Online]. Available: https://arxiv.org/abs/1911.11907

[56] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[57] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Sep. 1997.

[58] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, [Online]. Available: https://arxiv.org/abs/1706.03762.

[60] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," Hugging Face, Brooklyn, USA, 2020, [Online]. Available: https://arxiv.org/abs/1910.03771.

[61] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[62] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Pearson, 2023.

[63] J. Alammar, "The illustrated bert, elmo, and co. (how nlp cracked transfer learning)," Blog post, *jalammar.github.io*, 2018, [Online]. Available: https://jalammar.github.io/illustrated-bert/.

[64] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: Decoding-enhanced bert with disentangled attention," in *Proc. Int. Conf. on Learning Representations (ICLR)*, Virtual Event, 2021. [Online]. Available: https://arxiv.org/abs/2006.03654

[65] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019, [Online]. Available: https://arxiv.org/abs/1907.11692.

[66] P. He, J. Gao, and W. Chen, "Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing," *arXiv preprint arXiv:2111.09543*, 2021. [Online]. Available: https://arxiv.org/abs/2111.09543

[67] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," in *Proc. Int. Conf. on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2020, [Online]. Available: https://arxiv.org/abs/2003.10555.

[68] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[69] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.

[70] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," *arXiv preprint arXiv:1806.03822*, 2018.

[71] S. Mehta and M. Rastegari, "Separable self-attention for mobile vision transformers," 2022. [Online]. Available: https://arxiv.org/abs/2206.02680

[72] S. Mehta and M. Rastegari., "Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer," https://arxiv.org/abs/2110.02178, 2022.

[73] J. Bromley, J. Bentz, L. Bottou, I. Guyon, Y. Lecun, C. Moore, E. Sackinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, p. 25, 08 1993.

[74] Z. Zhou, Y. Yang, and Z. Li, "Apsn: Adversarial pseudo-siamese network for fake news stance detection," *Electronics*, vol. 12, no. 4, p. 1043, 2023. [Online]. Available: https://doi.org/10.3390/electronics12041043

[75] W. Jiang, S. Zhang, and K. He, "Siamese transformer networks for few-shot image classification," 2024. [Online]. Available: https://arxiv.org/abs/2408.01427

[76] M.-H. Guo, T. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu, "Attention mechanisms in computer vision: A survey," *Computational Visual Media*, vol. 8, pp. 331–368, 2022.

[77] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," 2018. [Online]. Available: https://arxiv.org/abs/1807.06521

[78] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," 2021. [Online]. Available: https://arxiv.org/abs/2103.02907

[79] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 1–9.

[80] M. Huddar, S. Sannakki, and V. Rajpurohit, "Multi-level context extraction and attention-based contextual inter-modal fusion for multimodal sentiment analysis and emotion classification," *International Journal of Multimedia Information Retrieval*, vol. 9, pp. 1–10, 06 2020.

[81] F. Zhao, C. Zhang, and B. Geng, "Deep multimodal data fusion," *ACM Computing Surveys*, vol. 56, no. 9, pp. 216:1–216:36, Apr. 2024. [Online]. Available: https://doi.org/10.1145/3649447

[82] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2019.

[83] H. Lin, C. Chen, and H. Shuai, "Multi-modal fake news detection using textual and visual features," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6.

[84] B. Duke and G. W. Taylor, "Generalized hadamard-product fusion operators for visual question answering," 2018. [Online]. Available: https://arxiv.org/abs/1803.09374

[85] S.-Y. Lin, Y.-C. Chen, Y.-H. Chang, S.-H. Lo, and K.-M. Chao, "Text–image multimodal fusion model for enhanced fake news detection," *Science Progress*, vol. 107, no. 4, pp. 1–31, 2024. [Online]. Available: https://journals.sagepub.com/doi/10.1177/00368504241292685

[86] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, 1998.

[87] H. Gunes and M. Piccardi, "Fusing face and body gesture for emotional recognition," in *Proc. Int. Conf. Image Process. (ICIP)*, vol. 2, 2005, pp. 306–311.

[88] D. Zhang, M. Islam, and G. Lu, "A review on automatic image annotation techniques," *Pattern Recognition*, vol. 45, no. 1, pp. 346–362, 2012, section on decision-level fusion discusses common rules such as maximum and average.

[89] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," 2017. [Online]. Available: https://arxiv.org/abs/1705.09406

[90] Y. Huang, "Multimodal fake news detection based on contrastive learning and cross-attention mechanism," in *Proceedings of the 2024 International Conference on Artificial Intelligence and Network Technology (AINIT)*, Mar. 2024, pp. 2219–2226.

[91] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," 2017. [Online]. Available: https://arxiv.org/abs/1606.00061

[92] D.-K. Nguyen and T. Okatani, "Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6087–6096.

[93] Y. Wu, P. Zhan, Y. Zhang, L. Wang, and Z. Xu, "Multimodal fusion with co-attention networks for fake news detection," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 2560–2569. [Online]. Available: https://aclanthology.org/2021.findings-acl.226/

[94] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," 2019. [Online]. Available: https://arxiv.org/abs/1908.02265

[95] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visualbert: A simple and performant baseline for vision and language," 2019. [Online]. Available: https://arxiv.org/abs/1908.03557

[96] Z. Jin, J. Cao, H. Guo, Y. Zhang, and J. Luo, "Multimodal fusion with recurrent neural networks for rumor detection on microblogs," in *Proc. 25th ACM Int. Conf. on Multimedia (MM)*. New York, NY, USA: ACM, Oct. 2017, pp. 795–816.

[97] F. Ma, *Information Communication*. Morgan and Claypool Publishers, Jul. 2015.

[98] Xinhua News Agency, "Official news agency of china," http://www.xinhuanet.com/, accessed: 2025-11-30.

[99] A. S. Tomar, A. Sharma, A. Shrivastava, A. S. Rana, and P. Yadav, "A comparative analysis of activation function, evaluating their accuracy and efficiency when applied to miscellaneous datasets," in *Proc. 2nd Int. Conf. Appl. Artif. Intell. Comput. (ICAAIC)*, 2023, pp. 1035–1042.

[100] S. Verma, A. Chug, A. Singh, and D. Singh, "Pds-mcnet: a hybrid framework using mobilenetv2 with silu6 activation function and capsule networks for disease severity estimation in plants," *Neural Computing and Applications*, vol. 35, pp. 1–24, 06 2023.

[101] M. Lee, "Gelu activation function in deep learning: A comprehensive mathematical analysis and performance," *arXiv preprint arXiv:2305.12073*, 2023. [Online]. Available: https://arxiv.org/abs/2305.12073

[102] Z. Vujovic, "Classification model evaluation metrics," *International Journal of Advanced Computer Science and Applications*, vol. Volume 12, pp. 599–606, 07 2021.

[103] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[104] M. E. Paoletti, J. M. Haut, X. Tao, J. Plaza, and A. Plaza, "Flop-reduction through memory allocations within cnn for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5938–5952, 2021.

[105] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[106] A. Thota, P. Tilak, S. Ahluwalia, and N. Lohia, "Fake news detection: A deep learning approach," *SMU Data Science Review*, vol. 1, no. 3, 2018. [Online]. Available: https://scholar.smu.edu/datasciencereview/vol1/iss3/10

[107] A. K. Jha, "Fake news challenge," Kaggle, 2017, [Accessed: Sept. 20, 2025]. [Online]. Available: https://www.kaggle.com/datasets/abhinavkrjha/fake-news-challenge

[108] P. M. Subhash, D. Gupta, S. Palaniswamy, and M. Venugopalan, "Fake news detection using deep learning and transformer-based model," in *Proc. 14th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, Delhi, India, Jul. 6–8 2023, pp. 1–6.

[109] C. Bisaillon, "Fake and real news dataset," 2018, [Online]. Available: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset.

[110] "Isot fake news dataset," University of Victoria, ISOT Lab, Nov. 2007, dataset. [Online]. Available: https://doi.org/10.23721/100/1478816

[111] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Netw.*, vol. 18, no. 5-6, pp. 602–610, Jul. 2005.

[112] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 20, no. 2, pp. 215–232, Jul. 1958.

[113] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[114] R. A. Kamble, V. Pareek, and T. Jain, "Fake news detection using machine learning and deep learning," in *Proceedings of the 2024 IEEE International Conference on Smart Power Control and Renewable Energy (ICSPCRE)*, Rourkela, India, 2024, pp. 1–6.

[115] P. K. Verma, P. Agrawal, L. Amorim, and R. Prodan, "Welfake: Word embedding over linguistic features for fake news detection," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 4, pp. 881–893, Aug. 2021.

[116] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, San Francisco, CA, USA, 2016, pp. 785–794.

[117] V. Coder, "Fake news classification (image)," Kaggle, Sep. 2021, [Accessed: Sept. 20, 2025]. [Online]. Available: https://www.kaggle.com/datasets/vivek1coder/fake-news-classification-image

[118] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 91–99.

[119] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, June 2012.

[120] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. European Conf. on Computer Vision (ECCV)*, 2014, pp. 740–755.

[121] N. I. of Standards and T. (NIST), "Nimble challenge 2016 evaluation dataset," [Online]. Available: https://www.nist.gov/itl/iad/mig/nimble-challenge-2016-evaluation-dataset, 2016.

[122] J. Dong, W. Wang, and T. Tan, "Casia image tampering detection evaluation database," in *Proc. IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, Jul. 2013, pp. 422–426.

[123] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, and G. Serra, "A sift-based forensic method for copy–move attack detection and transformation recovery," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1099–1110, 2011.

[124] Y. Hsu, S. Chang, and H. Chou, "Image splicing detection using camera response function consistency and automatic segmentation," *Proc. IEEE Int. Conf. Multimedia and Expo (ICME)*, pp. 28–31, 2007.

[125] S. Singhal, R. R. Shah, T. Chakraborty, P. Kumaraguru, and S. Satoh, "Spotfake: A multimodal framework for fake news detection," 09 2019, pp. 39–47.

[126] S. K. Hamed, M. J. A. Aziz, and M. R. Yaakub, "Improving data fusion for fake news detection: a hybrid fusion approach for unimodal and multimodal data," *IEEE Access*, vol. 12, pp. 112 412–112 425, 2024.

[127] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2017. [Online]. Available: https://arxiv.org/abs/1610.02357

[128] P. Singh, R. Srivastava, K. P. S. Rana, and V. Kumar, "Semi-fnd: Stacked ensemble based multimodal inferencing framework for faster fake news detection," *Expert Systems with Applications*, vol. 215, p. 119302, Mar. 2023.

[129] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," Salt Lake City, UT, USA, pp. 8697–8710, 2018.

[130] C. Song, N. Ning, Y. Zhang, and B. Wu, "A multimodal fake news detection model based on crossmodal attention residual and multichannel convolutional neural networks," *Information Processing & Management*, vol. 58, no. 1, p. 102437, Jan. 2021.

[131] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "Fakenewsnet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media," 2019. [Online]. Available: https://arxiv.org/abs/1809.01286

[132] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," 2019. [Online]. Available: https://arxiv.org/abs/1706.09516

[133] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and 9th Int. Joint Conf. Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992.

[134] X. Zhou, A. Mulay, E. Ferrara, and R. Zafarani, "Recovery: A multimodal repository for COVID-19 news credibility research," *CoRR*, vol. abs/2006.05557, 2020. [Online]. Available: https://arxiv.org/abs/2006.05557

[135] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[136] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

[137] A. Yadav, S. Gaba, H. Khan, I. Budhiraja, A. Singh, and K. K. Singh, "Etma: Efficient transformer-based multilevel attention framework for multimodal fake news detection," *IEEE Trans. Comput. Soc. Syst.*, vol. 11, no. 4, pp. 5015–5027, Aug. 2024.

[138] J. Ruvika, "Fake news detection dataset," Kaggle, 2017, [Accessed: Sept. 20, 2025]. [Online]. Available: https://www.kaggle.com/datasets/jruvika/fake-news-detection

[139] P. Pontes, "Fake news detection dataset," Kaggle, 2017, Link unavailable as of Sept. 20, 2025.

[140] M. Risdal, "Fake news dataset," Kaggle, 2017, [Accessed: Sept. 20, 2025]. [Online]. Available: https://www.kaggle.com/datasets/mrisdal/fake-news

[141] X. Shen, M. Huang, Z. Hu, S. Cai, and T. Zhou, "Multimodal fake news detection with contrastive learning and optimal transport," *Frontiers in Computer Science*, vol. 6, p. 1473457, Nov 2024.

[142] A. Zubiaga, M. Liakata, and R. Procter, "Learning reporting dynamics during breaking news for rumour detection in social media," *arXiv preprint arXiv:1610.07363*, 2016.

[143] K. Tian, G. Rao, X. Wang, M. Yu, J. Zhang, and L. Zhang, "CMFNThinker: A novel cross-source multi-modal fake news detection model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2025, p. 10889602.

[144] M. Grootendorst, "Keybert: Minimal keyword extraction with bert," [Online]. Available: https://doi.org/10.5281/zenodo.4461265, 2020.

[145] J. Zhang, R. Gan, J. Wang, Y. Zhang, L. Zhang, P. Yang, X. Gao, Z. Wu, X. Dong, J. He, J. Zhuo, Q. Yang, Y. Huang, X. Li, Y. Wu, J. Lu, X. Zhu, W. Chen, T. Han, K. Pan, R. Wang, H. Wang, X. Wu, Z. Zeng, and C. Chen, "Fengshenbang 1.0: Being the foundation of chinese cognitive intelligence," 2023. [Online]. Available: https://arxiv.org/abs/2209.02970

[146] Y. Li, H. He, J. Bai, and D. Wen, "MCFEND: A multi-source benchmark dataset for chinese fake news detection," in *Proc. ACM Web Conf. (WWW)*, 2024, pp. 4018–4027. [Online]. Available: https://doi.org/10.1145/3589334.3645385

[147] Y. Lu and N. Yao, "A fake news detection model using the integration of multimodal attention mechanism and residual convolutional network," *Scientific Reports*, vol. 15, no. 20544, 2025.

[148] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6202–6211.

[149] S. M. Yasir and H. Kim, "Lightweight deepfake detection based on multi-feature fusion," *Applied Sciences*, vol. 15, no. 4, p. 1954, 2025.

[150] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[151] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[152] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1–11.

[153] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A large-scale challenging dataset for deepfake forensics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3207–3216.

[154] H. Guo, L. Wang, B. Li, and Z. Guo, "Tinydf: Tiny and effective model for deepfake detection," in *Advanced Intelligent Computing Technology and Applications*, D.-S. Huang, W. Chen, Y. Pan, and H. Chen, Eds. Singapore: Springer Nature Singapore, 2025, pp. 247–256.

[155] B. Zi, M. Chang, J. Chen, X. Ma, and Y.-G. Jiang, "Wilddeepfake: A challenging real-world dataset for deepfake detection," in *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*. Association for Computing Machinery, 2020, pp. 2382–2390.

[156] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The deepfake detection challenge (dfdc) dataset," 2020. [Online]. Available: https://arxiv.org/abs/2006.07397

[157] L. Jiang, R. Li, W. Wu, C. Qian, and C. C. Loy, "Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2886–2895.

[158] Sandhya and A. Kashyap, "A light weight depthwise separable layer optimized cnn architecture for object-based forgery detection in surveillance videos," *The Computer Journal*, vol. 67, no. 6, pp. 2270–2285, 2024.

[159] Y. Xiao, J. Wu, and C. Ma, "Ga-ghostnet: A lightweight cnn model for identifying pests and diseases using a gated multi-scale coordinate attention mechanism," *Engineering Letters*, vol. 32, no. 7, pp. 1281–1290, 2024.

[160] S. Chen, S. Tan, B. Li, and J. Huang, "Automatic detection of object-based forgery in advanced video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 11, pp. 2138–2151, 2016.

[161] X. Wu, C. Zhan, Y. Lai, M.-M. Cheng, and J. Yang, "Ip102: A large-scale benchmark dataset for insect pest recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8787–8796.

[162] Mdsakibullah, "Jute pest dataset," Kaggle, 2023, [Accessed: Oct. 11, 2025]. [Online]. Available: https://www.kaggle.com/datasets/mdsakibullah/jute-pest-dataset

[163] P. S. Thakur, "Embrapa," Kaggle, 2019, [Accessed: Oct. 11, 2025]. [Online]. Available: https://www.kaggle.com/datasets/poornimasinghthakur/embrapa

[164] L. Sar, "Apple disease dataset," Kaggle, 2021, [Accessed: Oct. 11, 2025]. [Online]. Available: https://www.kaggle.com/datasets/ludehsar/apple-disease-dataset

[165] B. Alsinglawi, C. McCarthy, S. Webb, C. Fluke, and N. Saidy, "A lightweight large vision-language model for multimodal medical images," *arXiv preprint arXiv:2504.05575*, Apr. 2025.

[166] P. Liu and X. Wang, "A lightweight multi-modal emotion recognition network based on multi-task learning," in *2021 International Conference on Neuromorphic Computing (ICNC)*, 2021, pp. 368–372.

[167] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language Resources and Evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[168] X. Lou, P. Gao, M. Hofmann, Y. Jiang, S. Liu, and P. Desrosiers, "Lite-mdetr: A lightweight multi-modal detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 15 330–15 339.

[169] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg, "Referitgame: Referring to objects in photographs of natural scenes," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 787–798.

[170] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, "Modeling context in referring expressions," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 69–85.

[171] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy, "Generation and comprehension of unambiguous object descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 11–20.

[172] T. Wang, G. Chen, X. Zhang, C. Liu, J. Wang, X. Tan, W. Zhou, and C. He, "Lmfnet: Lightweight multimodal fusion network for high-resolution remote sensing image segmentation," *Pattern Recognition*, vol. 164, p. 111579, Aug. 2025. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2025.111579

[173] M. Bosch, A. Leichtman, J. Chilson, K. Foster, G. Christie, S. Wang, G. D. Hager, and M. Brown, "Semantic stereo for incidental satellite images," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 1524–1532, uS3D Dataset. [Online]. Available: https://ieeexplore.ieee.org/document/8659068

[174] W. G. I. ISPRS Commission II, "2d semantic labeling - potsdam," 2012, iSPRS 2D Semantic Labeling Benchmark Dataset. [Online]. Available: https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx

[175] ——, "2d semantic labeling - vaihingen," 2012, iSPRS 2D Semantic Labeling Benchmark Dataset. [Online]. Available: https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-vaihingen.aspx

[176] S. Zhang, Y. Xu, N. Usuyama, H. Xu, J. Bagga, R. Tinn, S. Preston, R. Rao, M. Wei, N. Valluri, C. Wong, A. Tupini, Y. Wang, M. Mazzola, S. Shukla, L. Liden, J. Gao, A. Crabtree, B. Piening, C. Bifulco, M. P. Lungren, T. Naumann, S. Wang, and H. Poon, "Biomedclip: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs," 2025. [Online]. Available: https://arxiv.org/abs/2303.00915

[177] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[178] Y. Hu, T. Li, Q. Lu, W. Shao, J. He, Y. Qiao, and P. Luo, "Omnimedvqa: A new large-scale comprehensive evaluation benchmark for medical lvlm," 06 2024, pp. 22 170–22 183.

[179] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning Workshop*, 2015.

[180] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *European Conference on Computer Vision Workshops*, 2016, pp. 850–865.

[181] B. N. Oreshkin, P. R. López, and A. Lacoste, "Tadam: Task dependent adaptive metric for improved few-shot learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 721–731. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/80cb9290b7fd50ce6bc3bee7b9f994c6-Abstract.html

[182] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017. [Online]. Available: https://openreview.net/forum?id=rJY0-Kcll

[183] L. Bertinetto, J. F. Henriques, P. H. S. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," in *Proceedings of the 7th International Conference on Learning Representations Workshop (ICLR Workshop)*, 2019. [Online]. Available: https://openreview.net/forum?id=HyxnZh0ct7

[184] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," in *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018. [Online]. Available: https://openreview.net/forum?id=HJcSzz-CZ

[185] S. Han, L. Gao, Y. Wu, T. Wei, M. Wang, and X. Cheng, "Siamsmn: Siamese cross-modality fusion network for object tracking," *Information*, vol. 15, no. 7, 2024. [Online]. Available: https://www.mdpi.com/2078-2489/15/7/418

[186] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015. [Online]. Available: https://arxiv.org/abs/1405.0312

[187] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," 2015. [Online]. Available: https://arxiv.org/abs/1409.0575

[188] H. Fan, H. Bai, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, Harshit, M. Huang, J. Liu, Y. Xu, C. Liao, L. Yuan, and H. Ling, "Lasot: A high-quality large-scale single object tracking benchmark," 2020. [Online]. Available: https://arxiv.org/abs/2009.03465

[189] J. Huertas-Tato, C. Koutlis, S. Papadopoulos, D. Camacho, and I. Kompatsiaris, "A clip-based siamese approach for meme classification," 2024. [Online]. Available: https://arxiv.org/abs/2409.05772

[190] C. Sharma, D. Bhageria, W. Scott, S. PYKL, A. Das, T. Chakraborty, V. Pulabaigari, and B. Gamback, "Semeval-2020 task 8: Memotion analysis – the visuo-lingual metaphor!" 2020. [Online]. Available: https://arxiv.org/abs/2008.03781

[191] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine, "The hateful memes challenge: Detecting hate speech in multimodal memes," 2021. [Online]. Available: https://arxiv.org/abs/2005.04790

[192] S. Pramanick, S. Sharma, D. Dimitrov, M. S. Akhtar, P. Nakov, and T. Chakraborty, "Momenta: A multimodal framework for detecting harmful memes and their targets," 2021. [Online]. Available: https://arxiv.org/abs/2109.05184

[193] S. Suryawanshi, B. R. Chakravarthi, M. Arcan, and P. Buitelaar, "Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text," in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*. European Language Resources Association (ELRA), 2020, pp. 32–41. [Online]. Available: https://aclanthology.org/2020.trac-1.6/

[194] D. Dimitrov, B. B. Ali, S. Shaar, F. Alam, F. Silvestri, H. Firooz, P. Nakov, and G. D. S. Martino, "Detecting propaganda techniques in memes," 2021. [Online]. Available: https://arxiv.org/abs/2109.08013

[195] P. Ye, G. Xiao, and J. Liu, "Multimodal features alignment for vision–language object tracking," *Remote Sensing*, vol. 16, no. 7, p. 1168, Mar. 2024.

[196] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome, "Mutan: Multimodal tucker fusion for visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2612–2620.

[197] Z. Li, R. Tao, E. Gavves, C. G. M. Snoek, and A. W. M. Smeulders, "Tracking by natural language specification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6495–6503.

[198] X. Wang, X. Shu, Z. Zhang, B. Jiang, Y. Wang, Y. Tian, and F. Wu, "Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual Conference, Jun. 2021, pp. 13 763–13 773.

[199] Z. Wan, P. Zhang, Y. Wang, S. Yong, S. Stepputtis, K. Sycara, and Y. Xie, "Sigma: Siamese mamba network for multi-modal semantic segmentation," in *Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV)*, 2025, pp. 1734–1744.

[200] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2024.

[201] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, J. Jiao, and Y. Liu, "Vmamba: Visual state space model," 2024. [Online]. Available: https://arxiv.org/abs/2401.10166

[202] Q. Ha, K. Watanabe, T. Karasawa, Y. Ushiku, and T. Harada, "Mfnet: Towards real-time semantic segmentation for autonomous vehicles with multispectral scenes," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017, pp. 5108–5115.

[203] S. S. Shivakumar, N. Rodrigues, A. Zhou, I. D. Miller, V. Kumar, and C. J. Taylor, "Pst900: Rgb-thermal calibration, dataset and segmentation network," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 9441–9447.

[204] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Proc. Eur. Conf. on Computer Vision (ECCV)*, 2012, pp. 746–760.

[205] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 567–576.

[206] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," 2017. [Online]. Available: https://arxiv.org/abs/1611.01578

[207] T.-J. Yang, A. Howard, B. Chen, X. Zhang, A. Go, M. Sandler, V. Sze, and H. Adam, "Netadapt: Platform-aware neural network adaptation for mobile applications," 2018. [Online]. Available: https://arxiv.org/abs/1804.03230

[208] M. Elsayed Abd Elaziz, M. Al-qaness, A. Dahou, S. Alsamhi, L. Abualigah, R. Ibrahim, and A. Ewees, "Evolution toward intelligent communications: Impact of deep learning applications on the future of 6g technology," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 14, 11 2023.

[209] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 2023. [Online]. Available: https://arxiv.org/abs/1606.08415

[210] K. Gupta, T. Ajanthan, A. van den Hengel, and S. Gould, "Understanding and improving the role of projection head in self-supervised learning," 2022. [Online]. Available: https://arxiv.org/abs/2212.11491

[211] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L 2 hypersphere embedding for face verification," in *Proceedings of the 25th ACM international conference on Multimedia*. ACM, Oct. 2017, p. 1041–1049. [Online]. Available: http://dx.doi.org/10.1145/3123266.3123359

[212] Z. Wang, Z. Liu, and Z. Hu, "Deep multimodal fusion networks," *IEEE Transactions on Multimedia*, 2020.

[213] J. Xue, Y. Wang, Y. Tian, Y. Li, L. Shi, and L. Wei, "Detecting fake news by exploring the consistency of multimodal data," *Information Processing Management*, vol. 58, p. 102610, 09 2021.

[214] Y. Shou, H. Liu, X. Cao, D. Meng, and B. Dong, "A low-rank matching attention based cross-modal feature fusion method for conversational emotion recognition," 2024. [Online]. Available: https://arxiv.org/abs/2306.17799

[215] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General perception with iterative attention," 2021. [Online]. Available: https://arxiv.org/abs/2103.03206

[216] M. S. Ryoo, A. Piergiovanni, A. Arnab, M. Dehghani, and A. Angelova, "Tokenlearner: What can 8 learned tokens do for images and videos?" 2022. [Online]. Available: https://arxiv.org/abs/2106.11297

[217] J. Lee, Y. Lee, J. Kim, A. R. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," 2019. [Online]. Available: https://arxiv.org/abs/1810.00825

[218] Z. Liu, Y. Shen, V. B. Lakshminarasimhan, P. P. Liang, A. Zadeh, and L.-P. Morency, "Efficient low-rank multimodal fusion with modality-specific factors," 2018. [Online]. Available: https://arxiv.org/abs/1806.00064

[219] E. Stevens, L. Antiga, and T. Viehmann, *Deep Learning with PyTorch*. O'Reilly Media, 2020.

[220] C. Shorten and T. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, 07 2019.

[221] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016. [Online]. Available: https://arxiv.org/abs/1609.08144

[222] S. F. Chaerul Haviana, S. Mulyono, and Badie'Ah, "The effects of stopwords, stemming, and lemmatization on pre-trained language models for text classification: A technical study," in *2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2023, pp. 521–527.

[223] H. Face, "Transformers: State-of-the-art machine learning for pytorch, tensorflow, and jax," https://huggingface.co/docs/transformers, 2024.

[224] R. Wightman, "Pytorch image models (timm)," https://github.com/huggingface/pytorch-image-models, 2024.

[225] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," 2020. [Online]. Available: https://arxiv.org/abs/1909.11942

[226] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020. [Online]. Available: https://arxiv.org/abs/1910.01108

[227] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019. [Online]. Available: https://arxiv.org/abs/1711.05101

[228] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," 2019. [Online]. Available: https://arxiv.org/abs/1907.10902