

University of Sheffield

A Multi-Modal Fake News Classifier using Transfer Learning



Leslie Canas

Supervisor: Emma Norling

A report submitted in partial fulfilment of the requirements
for the degree of MSc in Data Analytics

in the


Department of Computer Science

October 5, 2022

Declaration

All sentences or passages quoted in this document from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure.

Name: Leslie Canas

Signature: 

Date: 5th October 2022

Abstract

The way in which we consume news has considerably shifted away from traditional printed newspapers to online news sites due to its ease of access, lower costs for publishers, and the timeliness of information. However, through this medium of news dissemination, the scale and speed of false reporting has also escalated. Fake news has critical social, economic, and political impact as seen in the 2016 US Election and COVID-19 events. Manual fact-checking however, can be costly and insufficient to cover the amount of fake news circulated. Therefore, more automated solutions can help in this endeavour. Current fake news classifiers have largely focused on analysing the textual content and have ignored the images present in a news article.

This project aims to design, implement and evaluate a multi-modal fake news classifier to leverage both text and image in news articles using transfer learning methods. A modified model architecture proposed by Singhal et al. (2020) in their SpotFake+ model was implemented on the Gossipcop dataset. During this process, several experiments were conducted to evaluate choices in different Transformer-based models (BERT, RoBERTa and XLNET), convolutional neural networks (VGG and Xception), source task datasets (ImageNet and VGGFace), and level of fine-tuning for the pre-trained models.

This work has shown that transfer learning methods through feature extraction has some good results in textual features. However, both feature extraction and fine-tuning methods are unsuccessful with image features. In contrast to the results in SpotFake+ model but in line with the conclusions of Al Obaid et al. (2022), this work reveals that the usefulness of images in a multi-modal fake news classifier is dependent on the dataset. In addition, when leveraging transfer learning methods, researchers must pay attention to their choices of pre-trained models, source task datasets, and degree of fine-tuning as these have an effect on a model’s performance. Several improvements are suggested to increase the performance of the model but society can further benefit from an explainable fake news classifier model to understand the characteristics of a fake news article.

Acknowledgements

To my supportive husband, Tom, thank you for your sacrifices over the past year. You continued to push me to excel and be the best version of myself. Thank you for being a good sounding board when I had doubts during my research process.

I would also like to dedicate this to my father, Eduardo, who always encouraged me to read widely and ask questions. You emphasised the value of continuous learning which I will always take with me. I have come along way in my academic pursuits because of what you imparted.

To my supervisor, Emma Norling, thank you for providing guidance on the project and allowing me to pursue my own academic interest.

Finally, I would like to express my gratitude to the Fretwell-Downing Scholarship which sponsored my Master's degree. The financial support truly made the pursuit of this degree possible.

Contents

1	Introduction	1
1.1	Aims and Objectives	2
1.2	Overview of the Report	3
2	Literature Review	4
2.1	Current Approaches in Multi-Modal Fake News Classification	4
2.2	Transfer Learning	7
2.3	Vector Representation of Text	9
2.4	Attention Mechanism	10
2.5	Transformers	11
2.5.1	Bidirectional Encoder Representations from Transformers (BERT) . .	12
2.5.2	A Robustly Optimized BERT Pretraining Approach (RoBERTa) . . .	12
2.5.3	XLNET	13
2.6	Convolutional Neural Networks (CNNs)	13
2.6.1	Visual Geometry Group (VGG)	13
2.6.2	Xception	14
2.7	Summary	15
3	Methodology	16
3.1	Dataset	16
3.2	Proposed Model Architecture	17
3.3	Text Features	18
3.4	Image Features	19
3.5	Data Splitting	19
3.6	Weight Balancing During Training	19
3.7	Metrics	20
3.8	Summary	20
4	Experiments and Results	21
4.1	Experiment 1: Text Embeddings	21
4.2	Experiment 2 : Image Embeddings Without Fine Tuning	23
4.3	Experiment 3 : Image Embeddings From Different Source Task Datasets . . .	24
4.4	Experiment 4: Effect of Fine-Tuning on the Best Performing Image Model . .	24
4.5	Experiment 5: Combining the Best Performing Text and Image Models . . .	27
4.6	Summary	28

5	Discussion	29
5.1	Experiment 1: Textual Embeddings	29
5.2	Experiment 2: Image Embeddings Without Fine-tuning	30
5.3	Experiment 3: Image Embeddings From Different Source Task Datasets . . .	30
5.4	Experiment 4: Effect of Fine-Tuning On The Best Image Model	31
5.5	Experiment 5: Multi-modal Fake News Classifier	31
5.6	Challenges in Multi-Modal Fake News Classifiers	32
6	Conclusions	33
	Appendices	39
A	Monitoring Training	40

List of Figures

1.1	Multi-modal news samples	2
2.1	Network based deep transfer learning (Plested & Gedeon 2022)	9
2.2	Basic Encoder-Decoder Structure	10
2.3	The Encoder Part of the Transformer Architecture (Vaswani et al. 2017) . . .	12
2.4	Inception Module	14
2.5	Xception Module	15
3.1	Proposed Architecture	18
4.1	Architecture of Text Only Model	21
4.2	Architecture of Image Only Model	23
4.3	The Xception Architecture (Chollet 2016)	25
4.4	Confusion Matrix of the Multi-Modal Model	28
A.1	Monitoring Training on the BERT Model	40
A.2	Monitoring Training on the RoBERTa Model	40
A.3	Monitoring Training on the XLNET Model	41
A.4	Monitoring Training on the VGG16 Model	41
A.5	Monitoring Training on the VGG19 Model	41
A.6	Monitoring Training on the Xception Without Fine Tuning Model	41
A.7	Monitoring Training on the VGGFace 16 Model	42
A.8	Monitoring Training on the Different Levels of Fine Tuning on Xception Network	42
A.9	Monitoring Training on the Multi-Modal Model	43

List of Tables

3.1	Number of samples in the Gossipcop dataset before and after data preparation	17
4.1	Comparing different textual embeddings	22
4.2	Comparing different image embeddings	23
4.3	Comparing different source dataset for image embeddings	24
4.4	Comparing different amounts of fine tuning	26
4.5	Evaluation of Multi-Modal Classifier	27

Chapter 1

Introduction

Technology has changed the way society consume and disseminate news. There has been a shift away from traditional printed newspapers and movement towards newspaper websites or social media platforms. In a recent Ofcom study to understand the news consumption trends in the UK, there has been a 57 percent decline in the circulation volume of national newspaper titles between 2010 and 2019 (Ofcom 2020). However the ease of access, low cost, and velocity of news through the internet comes with a cost.

The presence of false news or in contemporary terms “fake news” in society goes as far back as you intend to look. Technology however, has accelerated the spread and escalated the impact of false reporting. Initially, the term *fake news* was largely associated with the political domain such that it gained popularity during the 2016 U.S. Presidential Election; in three years, it also influenced the health domain during the COVID-19 pandemic. The dissemination of false information has been noted to deteriorate the status of democracy, undermine public opinion formation, and increase distrust in media (Giusti & Piras 2021, Chambers 2020).

There is no unified definition of the term “fake news” as it can appear in a multitude of forms such as rumors, satire, propaganda, and clickbait. For the purposes of this project, a wider definition is employed and taken from Jaster and Lanius (2021, p.20) where “fake news is news that lacks truth and truthfulness. It lacks truth in the sense that is either literally false or communicates something false ... with the intention to deceive or without concern for the truth”. Such definition captures information that is completely or partially false. Given the speed, scale, and proliferation of online news, the manual task of verifying the authenticity of news through fact checkers is a challenge that requires the help of more automated solutions.

Majority of the research on classifying fake news has largely been focused on understanding the textual content of the news. However, most online news contain multiple modalities to complement the textual content, such as images or videos, in order to grab the readers’ attention. Hence, there is a need to further develop models that are able to process multi-modal news content. Figure 1.1 displays a sample each of fake and real news containing texts and images.

Initial solutions in developing fake news classifiers relied on manually engineering features that are fed into classical machine learning algorithms such as Support Vector Machines or Gradient Boosted Trees. Developing rule-based or feature-based models can be time

Malia Obama Arrested With A Gang Of Thugs In Chicago



POSTED BY: BREAKING13NEWS



Malia Obama may have done irreparable harm to her career this morning when she decided to join a gang of thugs in Chicago for a day of drinking, drugs and dogfighting at a public park in Chicago. Malia was arrested along with seven others and charged with wanton endangerment of animals, public intoxication and possession of a controlled substance.

(a) Fake News

Fyre Festival 'Postponed' Amid Reports of 'Chaos' in the Bahamas

"It's every man for himself," one 21-year-old attendee told NBC News.



Fyre Festival attendees gather with their belongings outside event headquarters to wait for information on April 28, 2017 in Exumas, Bahamas.



April 28, 2017, 6:47 PM EDT / Updated April 28, 2017, 6:47 PM EDT

By Daniel Arkin and Shannon Wallers

It was supposed to be a "once-in-a-lifetime musical experience" on a remote island in the Bahamas. The organizers of the much-hyped Fyre Festival promised "two transformative weekends" of Instagram-ready luxury — world-class cuisine, private jets, yachts.

Sponsored Stories



(b) Real News

Figure 1.1: Multi-modal news samples

consuming and could require expertise in various fields such as linguistics and computer vision. The advancement of deep learning based solutions in both Natural Language Processing (NLP) and computer vision has seen a shift away from machine learning models in the field as it no longer requires feature engineering.

1.1 Aims and Objectives

Fake news detection is a classification task. As such the aim of this project is to design, implement, and evaluate a multi-modal fake news classification model using transfer learning methods applied to full length news articles. The modalities used are text and images which are sourced from a publicly made available repository called FakeNewsNet which is comprised of two datasets - Politifact and Gossipcop. It is one of the latest datasets for fake news research that contains multimodal data. However due to the limiting size of the Politifact dataset, the project specifically focused on the Gossipcop dataset.

In selecting the textual representations, the performance of three widely used Transformer-based models: BERT, RoBERTa, and XLNet, will be compared. For the visual representations, the performance of features extracted from deep learning networks such as VGG16, VGG19, and Xception, with pre-trained weights on different source task data will be evaluated. As the image modality has been less researched in the field, further experiments to leverage transfer learning techniques will be implemented. This includes using a different two source task datasets - ImageNet and VGGFace. Furthermore, after selecting the best performing deep learning architecture for representing the images, the level of fine-tuning of the best pre-trained model for feature extraction will be experimented. This will give some insight on the effectiveness of transfer learning methods.

The model architecture applied in this project is similar to the proposed SpotFake+ model where the textual and visual features are concatenated and then passed through several

dense layers before the classification layer (Singhal et al. 2020). The focus of this project is to understand the effect of the different textual and visual representations and transfer learning methods rather than proposing a novel model architecture. Unlike previous works, this project firstly evaluates the different textual and visual representations before building a model.

1.2 Overview of the Report

This chapter presented the background and purpose of this project. The next chapter contains a literature review of similar works that aimed to develop a multi-modal fake news classifier. Through this review, suitable methods and approaches can be derived and gaps in the research can be identified. The chapter also presents the building blocks of the proposed model which includes Transformer-based networks, deep convolutional networks, and transfer learning techniques.

Chapter 3 details the methodology of the project which covers the dataset used, the preprocessing steps, the proposed model architecture and the detailed implementation of the model's components.

Chapter 4 describes the experimental design and processes implemented. It also presents the results obtained from the different experiments conducted.

Chapter 5 provides a discussion around the results of all the experiments.

Chapter 6 concludes the project and summarizes the main findings. Most importantly, improvements for future work are discussed.

Chapter 2

Literature Review

Fake news detection has been approached in different ways, from analysing the news content to determine its authenticity, detecting conflicting stances from social media opinions, and propagation based network analysis (Ansar & Goswami 2021). This project focuses on analysing the news content rather than the social content or the propagation of the news piece as it is crucial to detect fake news in the first instance, before it is spread across social media. The dominant approach in this research area only utilizes the textual features of the news content and relies heavily on hand-crafted features which can require domain expertise and a large amount of human effort. More recently, researchers have been combining other modalities with text such as images and videos in large part due to the availability of multi-modal training datasets such as TI-CNN, FakeNewsNet and NewsBag (Yang et al. 2018, Shu et al. 2018, Jindal et al. 2020).

In this chapter, recent advancements in multi-modal fake news field is presented as well as the deep learning methods and techniques used in this project such as transfer learning, Transformer-based models, and deep convolutional neural networks.

2.1 Current Approaches in Multi-Modal Fake News Classification

The increasing success of deep learning approaches in both computer vision and natural language processing has led to a large shift in research focus, moving away from machine learning to deep learning solutions such as Convolution Neural Networks (CNN), Recurrent Neural Networks, and Generative Adversarial Networks (GANS) (Hangloo & Arora 2022). Deep learning approaches have outperformed traditional machine learning models as the latter relies on hand-crafted features which often give rise to high-dimensional representations of the content. In contrast, deep learning models are able to extract the most relevant features automatically (Mridha et al. 2021). Deep learning models are known to perform better with the availability of substantial data. In order to curate a large fake news dataset, labels need to be manually applied which can be laborious and expensive. Due to the limited size of multi-modal datasets, transfer learning and a number of pre-trained models have been widely applied to this research area. Within multi-modal fake news research, this technique has been used by Giachanou et al. (2020), Singhal et al. (2019), and Cui, Wang & Lee (2019) to name a few.

Giachanou et al. (2020) proposed a model that is able to leverage several aspects in a news article, which includes multiple images, the full textual content and a similarity measure between the textual and visual features. The pre-trained models used for text and images are BERT and VGG16, respectively. VGG16 was pre-trained on the ImageNet dataset and the last convolutional layer of the network was used as a feature extractor. To combine all three aspects, concatenation was employed and this combined vector was fused with attention mechanism. Their proposed model was tested on the FakeNewsNet repository, specifically GossipCop. Several experiments were ran but the most pertinent experiment relates to whether utilizing multiple modalities can improve a fake news classifier model. When testing the performance between a model that only utilize one modality, either solely text or one image, the best performing model is the text only model with an F1-score of 76.28%. The one image model only achieved an F1-score of 36.78%. However, their experiment showed that combining modalities can improve the model as combining the text and one image result in a 2.61% improvement.

The usefulness of image features in a fake news classifier was also confirmed in the SpotFake model which was presented by Singhal et al. (2019). In their model, they extract the text features from a pre-trained BERT model and the image features were sourced from the second to the last layer of a VGG-19 model pre-trained on ImageNet. Their proposed architecture contains a simple concatenation of the two different features after being passed on to fully connected dense layers separately. Post the fusion of the features, another dense layer is added before the final classification layer. They tested their model on Twitter and Weibo datasets where they found that combining text and image features do improve their model's performance. On both datasets, the multi-modal model improved the text only model by 25%. Singhal et al. (2019) extended their SpotFake model by applying it to longer text representations such as those found in the FakeNewsNet dataset as well as using XLNET rather than BERT to extract text features. They also increased the number of fully connected dense layers after each feature and post the fusion of the features. The number of nodes on the dense layers after the text feature are 1000, 500, and 100 whilst there are 2000, 2000, 1000 and 100 nodes on the dense layers after the image feature. Post the concatenation of the features, there are 200, 100 and 50 nodes in the dense layers. They called this model SpotFake+. Their multi-modal architecture achieved an accuracy of 85.6% on the Gossipcop dataset, which was a 2% improvement on their text only model.

The results in SpotFake+ have been challenged in the work of Al Obaid et al. (2022). In comparing the performance of their multi-modal ensemble fake news model with the current state-of-the-art models, they attempted to reproduce the SpotFake+ model on the FakeNewsNet data. They reported an 85.23% accuracy score on the Gossipcop dataset which is close to the reported performance in Singhal et al. (2020). However, when reporting the macro-F1 score it only achieved 49.76% which indicates that the SpotFake+ model is not able to classify fake news well. The high accuracy score but low macro-F1 score can be explained by the lack of methods in the SpotFake+ model to deal with the imbalanced nature of the dataset (Al Obaid et al. 2022). With regards to the usefulness of images, the results of their ensemble model concluded that it is dependent on the dataset such that they found an improvement over the text only model when incorporating images on the Politifact dataset but for the Gossipcop dataset, the performances between the text only model and the multi-modal model were negligible.

Another multi-modal fake news model, called SAME, which was proposed by Cui, Wang & Lee (2019) focused on the impact of sentiment analysis on their model. Albeit a different focus, their experiment shed light on the importance of selecting an appropriate textual representation. Their model used a pre-trained GloVe embedding to extract the textual features from the news article. For the visual features, the first seven layers of a pre-trained VGG neural network (the number of layers was unspecified) on ImageNet was utilized. In their top performing model, following an 80-20 training and test split, their ablation study showed that the text features had a negative attribution to their model by around 2.6%. This study highlights that non-contextual word embeddings such as GloVe may be insufficient for the fake news classification task.

From the three works discussed above, there has been no rigorous effort to test which pre-trained models to extract the textual or image features lead to better performances. In the work of Raj & Meel (2021), they performed a comparative analysis of eight CNN architectures on three fake news datasets: TI-CNN, EMERGENT, and MICC-F220. In testing on different datasets, a number of models consistently outperformed and these are VGG16 and Xception. The VGG16 and VGG19 model architectures outperformed in the TI-CNN data. For the EMERGENT dataset, ResNet50 and Xception achieved higher accuracy scores whilst Xception and VGG16 performed best on the MICC-F220 dataset. Their work showed that there is variability in performances of different CNN models depending on the dataset in question. The most interesting observation from their study is that CNN models perform best when it consists of tampered images, as found in the MICC-F220 dataset. This observation was also confirmed by Pashine et al. (2021) where they compared different CNN architectures such as VGG19 and Xception on a dataset with DeepFake images and videos. Both networks performed well but the Xception network achieved a 2.7% higher F1 score at 76% compared with the VGG19 network.

In understanding the most effective pre-trained language model for textual features, Khan et al. (2021) explored a number of text representation such as BERT, RoBERTa, DistilBERT, ELECTRA, and ELMo. They evaluated all the different representations on three diverse datasets - LIAR, Fake or real news, and their own curated dataset with over 80,000 news articles across a wide range of domains. To extract the text representations, a single linear layer is appended on the top of the pre-trained language model. Their experiment revealed that Transformer-based language models such as BERT and its other variants perform better than the bi-directional LSTM-based language model, ELMo. Most importantly, their results showed that there is a positive relationship between the performance of Transformer-based language models and the size of their pre-trained parameters. The RoBERTa model consists of 125M parameters which achieved a 96% accuracy on the custom dataset whilst the smallest Transformer model, DistillBERT, achieved 93% on the same dataset.

Gaps Identified

When extracting the features for both text and images, there is a lack of research on which pre-trained models to leverage. For textual features, only BERT and XLNET have been explored. For the visual features, more pre-trained deep learning models have been experimented on however, there is an absence of work on whether fine-tuning pre-trained models can improve performance. For instance, Giachanou et al. (2020) takes the last convolutional layer of VGG16 network, Singhal et al. (2020) utilizes the second to the last convolutional layer of

the VGG19 network, and Cui, Wang & Lee (2019) extracted features from the first seven layers of VGG. The fake news research area can benefit from understanding which text and image features work best.

With regards to transfer learning techniques, particularly the source task dataset for visual representations, ImageNet has been the default source task dataset. ImageNet is a large scale visual database that contains over 14M images of nouns based from WordNet (Deng et al. 2009). This database has accelerated the field of computer vision. However, it can be viewed that the ImageNet dataset does not contain much similarity with the FakeNewsNet images, which largely consists of people rather than objects. It may be beneficial to experiment on a different source task dataset that contains people to see if this affects the image classification part of the model. Parkhi et al. (2015) curated an image dataset called, VGGFace, composed of 2.6M images of celebrities and trained on the face recognition task. Such source data is more closely related to the dataset in this project, Gossipcop.

It is also unclear whether fine-tuning a number of layers in a pre-trained model can be beneficial if the source and target datasets are not similar. None of the works reviewed experimented on fine-tuning the pre-trained models for neither the text nor the visual features. As such, this project will also explore varying the levels of fine-tuning.

2.2 Transfer Learning

The concept of transfer learning is applying knowledge gained from one task to another task. The unified definition of transfer learning is given by (Pan & Yang 2010, p.1347) as follows:

”Given a source domain D_S and learning task T_S , a target domain D_T and learning task T_T , transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S , where $D_S \neq D_T$, or $T_S \neq T_T$. ”

In addition, a domain (D) is made up of a feature space \mathcal{X} and a marginal probability distribution $P(X)$ over \mathcal{X} , where $X = x_1, \dots, x_n \in \mathcal{X}$. A task (T) comprises of a label space \mathcal{Y} and a conditional probability distribution $P(Y|X)$ which is learned from the training data pairs $\{x_i, y_i\}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. Therefore in technical terms, the goal of transfer learning is to learn the target conditional probability distribution $P(Y_T|X_T)$ in D_T with the knowledge gained from D_S and T_S , where $D_S \neq D_T$, or $T_S \neq T_T$.

Sequential Transfer Learning

There are different categories within transfer learning; it is classified based on whether the source and target task are the same, whether the source and target domains are the same, and the order of learning. In the multi-modal fake news research area, sequential transfer learning has been largely applied. In this type of transfer learning, the source and target task are different such that $T_S \neq T_T$, and the two tasks are learned one after another. (Ruder 2019)

Sequential transfer learning has a pre-training and an adaptation phase. During the pre-training phase, a model is trained on the source task, T_S , which is identified as the pre-trained model. Next in the adaptation phase, the knowledge gained from the pre-trained model is

transferred to a downstream task, T_T , with the aim of improving the performance of the model trained on the target task. It is also important to understand the theoretical benefits of sequential transfer learning. Erhan et al. (2010) postulates that pre-training a model acts as an implicit regularizer by initializing parameters that imposes constraints on the minima in which an objective function can optimize. This hypothesis was supported in the experiment of Erhan et al. (2010) using the InfiniteMNIST dataset. Their results showed that pre-training helps in the optimization process as it defines the starting point of the training process and subsequently restricts training to a subset of parameters that are useful for unsupervised learning.

To successfully leverage sequential transfer learning, the source task dataset must have some similarity with the target task dataset (Plested & Gedeon 2022). If the source and target datasets are not well related, negative transfer can occur which is when transferring knowledge from the source task negatively impacts the performance on the target task. (n.d.) highlights that the main root cause of negative transfer is the divergence in the joint distribution of the source domain $P_S(X, Y)$ and the target domain $P_T(X, Y)$. The higher the similarity in their joint distributions, the more valuable knowledge exists in the source domain that can be leveraged to improve the performance in the target domain.

Image Classification

In the context of an image classification task, transfer learning is applied to the target model by initialising weights (W) that are trained on the source task T_S from the source domain D_S (Plested & Gedeon 2022). Some or all of the weights from a pre-trained model can be retrained and subsequently transferred to the target domain D_T and target task T_T . The idea behind this is that if the source task dataset is significantly larger than the target task dataset, the low-level features that determine the basic structure of an image can be learned from the source task. More importantly, the low-level features learned can be transferred to the target data to accelerate the learning process and reduce the generalization error on the smaller dataset (Goodfellow et al. 2016).

In computer vision, the most common pre-trained models involve the use of convolutional layers. Layers furthest from the fully connected layers capture the global features whilst layers closest to the fully connected layers, which act as the classification layers, extract more task-specific features of the input (Goodfellow et al. 2016). From Figure 2.1 (Plested & Gedeon 2022), it can be seen that there are several ways in which a pre-trained model can be adapted to the target task. One strategy is to leverage the entire pre-trained model's convolutional layers as a feature extractor and only replace the classification layer for the target task. This means that during the adaptation phase, the convolutional layer weights of the pre-trained model are frozen. Another strategy, called fine-tuning, involves selectively freezing certain layers in the convolutional network and re-training the other layers with the target dataset. Empirically, transferring more layers from the pre-trained model result in better performances on the target task when the source and target datasets are more similar Plested & Gedeon (2022).

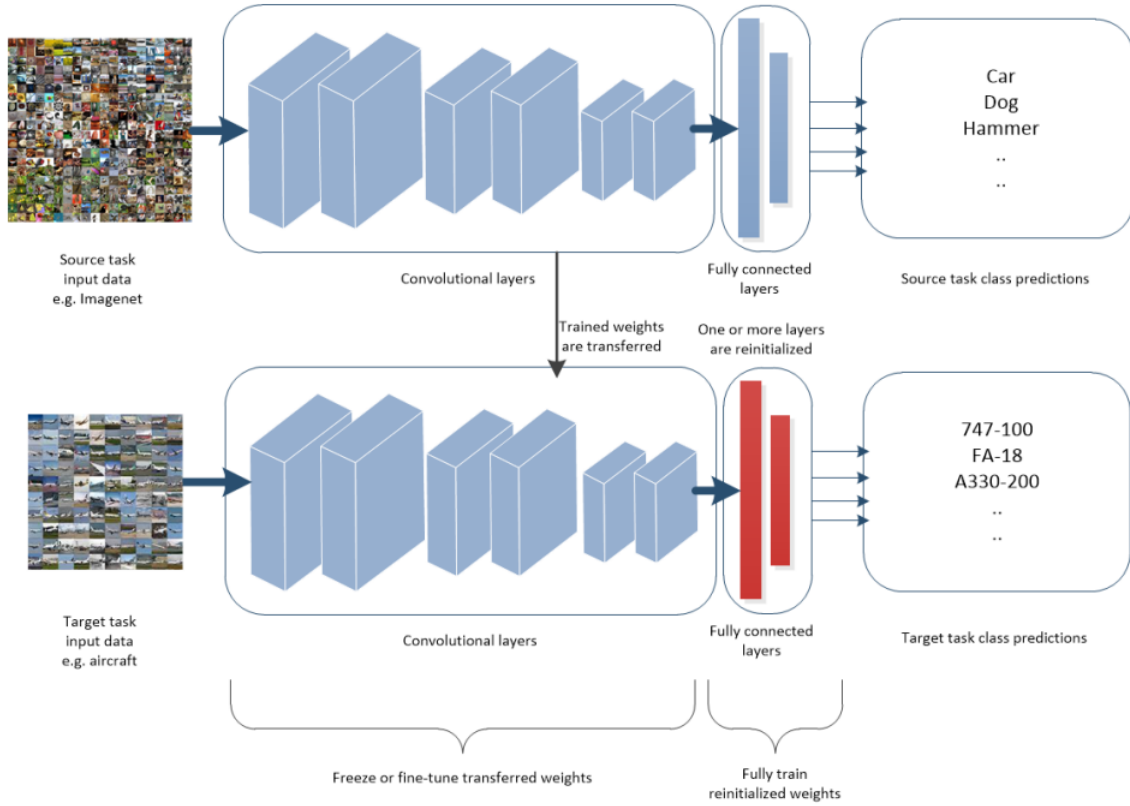


Figure 2.1: Network based deep transfer learning (Plested & Gedeon 2022)

Text Classification

In NLP, sequential transfer learning has been used in learning universal language representations. Howard & Ruder (2018) sets up the problem and aim as follows: given a source task, T_S and a target task T_T , where $T_S \neq T_T$, the goal is to boost the performance on T_T . They identified that the ideal source task is language modeling as it is able to capture several aspects in language that is transferable to other tasks such as long-term dependencies, hierarchical relations and sentiment. A good language representation is able to capture the lexical meaning, syntactic structures, and context of words. Howard & Ruder (2018) presented ULMFiT which is an effective transfer learning method that leveraged LSTM networks which were pre-trained on a large scale corpus and fine-tuned on several text classification tasks. Their method paved the way for pre-trained models in NLP as it outperformed state-of-the-art text classification models at the time.

2.3 Vector Representation of Text

The success of machine learning in NLP is dependent on providing algorithms a good language representation which are used as features. Techniques to capture the meaning of a particular

text has come along way from vector space models that stemmed from information retrieval systems, to word embeddings such as Word2vec and GloVe, and to the current dominant approach of contextualised representations. Contextualised representations made significant improvements to the static word embeddings as they are able to capture the semantic and syntactic properties of a word as it is used in a particular context (Pilehvar & Camacho-Collados 2020).

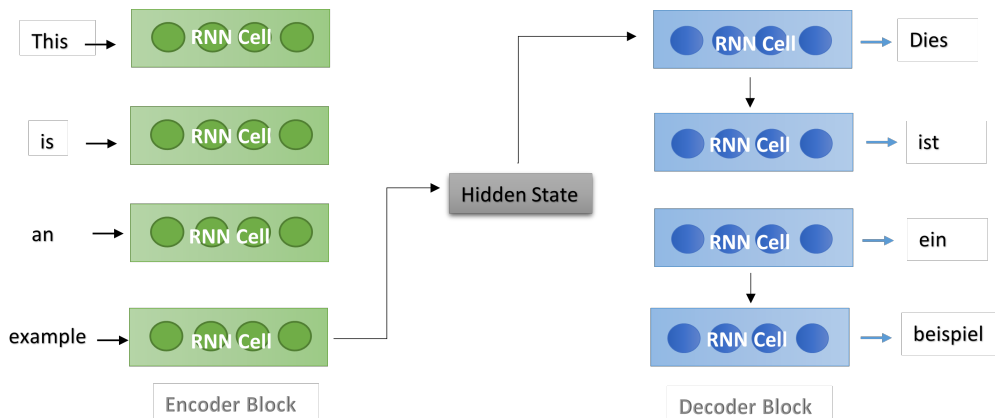


Figure 2.2: *Basic Encoder-Decoder Structure*

ULMFiT (Howard & Ruder 2018) is the work that started off the use contextual representations where the embeddings are learned from a pre-training a language model. By only leveraging LSTM, a type of recurrent architecture, long sequences still remain challenging. In the context of sequence to sequence learning, Cho et al. (2014) demonstrated that the performance of an LSTM based model rapidly falls as the length of the input sequence increases. This can be demonstrated in Figure 2.2 where the first RNN cell receives an input and outputs a representation of this input called a hidden state. As the network moves at each time step, the current hidden state is passed into the next input, until the network passes through all the inputs. However due to the sequential nature of the architecture, an information bottleneck in the last hidden state of the encoder occurs as it has to represent the meaning of the full input sequence which is then compressed to a fixed dimensional vector.

2.4 Attention Mechanism

Bahdanau et al. (2014) proposed a solution to the information bottleneck found in the early encoder-decoder architecture. Instead of the decoder only able to access the last hidden state from the encoder, the encoder gives a representation of all the inputs in each time step which the decoder can access. To illustrate the process performed by the encoder, given an input sequence of length T , the encoder calculates an annotation, h_i , for the entire input sequence, $\{x_0, x_1, \dots, x_T\}$ where x is a given word. Bahdanau et al. (2014) employed a bidirectional RNN in order for the annotation to summarize both the preceding and the following words. Therefore the annotation for each word, x_i , is represented as $h_t = [h_t^{\rightarrow}, h_t^{\leftarrow}]$ where h_t^{\rightarrow} is the forward hidden state and h_t^{\leftarrow} is the backward hidden state. The combined hidden states, h_t is passed to the decoder.

As h_t can be a large input for the decoder, an attention mechanism is put in place to allow the decoder to focus on the most relevant information. This is achieved by another component, in addition to the hidden state that is fed to the decoder, namely a context vector c which is computed as follows:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \quad (2.1)$$

The context vector is the weighted sum of the annotation h_t . The weighting component, α_{ij} or the amount attention given is defined as the softmax over the attention scores, e :

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \quad (2.2)$$

$$e_{ij} = a(s_{i-1}, h_j) \quad (2.3)$$

The attention score quantifies how much the inputs around position j and the output at position i align with each other. It takes the annotation, h_j , and the decoder output at the previous time step, s_{t-1} .

2.5 Transformers

Since the attention mechanism proposed by Bahdanau et al. (2014), different types of attention mechanisms have been proposed. Researchers from Google presented a novel sequence modelling neural network architecture called, *Transformer* (Vaswani et al. 2017). It follows an encoder-decoder architecture that is based solely on attention mechanisms, without any need for recurrence. In this subsection, the encoder component of the Transformer is addressed.

Vaswani et al. (2017) applied a specific attention mechanism called scaled-product dot attention. The process includes computing three vectors: query (Q), key (K), and value (V) for each input x_i . This is achieved by multiplying the input vector, x_i , with its respective weight matrices (W^Q, W^K, W^V). When computing the similarity score, to measure how much the Q and K vectors align with each other, matrix multiplication of these two vectors are performed as shown in Equation 2.4 (Vaswani et al. 2017). It is then scaled by the dimension of the key vector, $\sqrt{d_k}$, to prevent large values from the matrix multiplication operation. These values are the weights applied to the value vector, V .

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.4)$$

Building on this singular attention function, Vaswani et al. (2017) put forward a multi-head attention mechanism. It allows the attention mechanism to be performed multiple times, in a parallel and independent manner. Through the multi-head mechanism, the model uses more than one set of weight matrices to produce different Q , K , and V vectors for the same word. All these attention calculations are then concatenated that results in one final value:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \quad (2.5)$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$

The projections in the attention head are parameter matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ and $W_i^O \in \mathbb{R}^{d_{hd} \times d_{model}}$. Finally, the output of the multi-head attention layer is fed to a fully connected feed-forward sub-layer which is composed of two linear transformations using ReLU as their activation function (Vaswani et al. 2017).

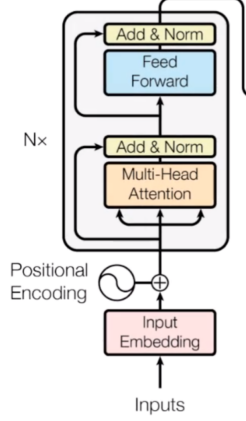


Figure 2.3: *The Encoder Part of the Transformer Architecture (Vaswani et al. 2017)*

This is where the power of Transformers comes in for language representation. The attention mechanism combined with multiple heads enables the model to attend to different parts of the input vector from multiple representation sub-spaces at various positions. For example, the phrase "dog chased rabbit" will have a different representation to "rabbit chased dog". To capture the position of the input in a sequence, in addition to receiving an input embedding, the Transformer model adds a positional encoding to the inputs which allows the model to take into account word order.

2.5.1 Bidirectional Encoder Representations from Transformers (BERT)

BERT is a pre-trained model which makes use of the encoder part of the Transformer architecture. Similar to ULMFiT, BERT incorporates pre-training language models on a large corpus, in their case they utilised the BookCorpus and Wikipedia (Devlin et al. 2018). It is able to incorporate context from both directions through one of its pre-training objective, Masked Language Modeling (MLM). In MLM, the model randomly masks 15% of the input embedding tokens and the goal is to predict these masked tokens. In order to handle text inputs, BERT uses WordPiece embeddings to tokenize a sequence. In addition, it makes use of a special classification token, [CLS], which is placed at the beginning of a sequence. In terms of its architecture, the base BERT model contains 12 attention heads, 12 layers with 768 dimension which equates to 110M learnable parameters.

2.5.2 A Robustly Optimized BERT Pretraining Approach (RoBERTa)

Liu et al. (2019) asserted that BERT was not sufficiently trained and further training of BERT can improve its performance. RoBERTa builds on the BERT model by training the model for a longer time period, with bigger data batches, with longer sequences and ten

times more data. The pre-training data included the Common Crawl dataset which contains 63M English news content. During pre-training, instead of applying a static mask on the token, the masked positions are dynamically changed. Their proposed training approach has outperformed BERT on several downstream tasks.

2.5.3 XLNET

As BERT is an autoencoder language model that is able to reconstruct the original sequence from the masked tokens, it assumes independence between the masked tokens. However, such assumption may not always hold true. Yang et al. (2019) improves on the MLM pre-training process found in BERT through permutation language modelling which are trained to predict a token based on the preceding tokens. However, as the model leverages the permutation operation, given an input sequence, each token will utilize the information from all positions. Such operation also compels the model to learn the dependencies across all token combinations and not just the context of the previous token. Yang et al. (2019) tested their architecture on a few text classification datasets and found that it outperforms BERT and RoBERTa on most of them.

2.6 Convolutional Neural Networks (CNNs)

CNNs have been used in image recognition for its ability to extract local features by restricting the receptive fields of a network's hidden layers locally. This is founded on the notion that image pixels that are next to each other are highly correlated (LeCun et al. 1999). CNN is a type of neural network that employs an operation called convolution, which is a linear operation that entails the multiplication of an input and a set of weights. Given a two-dimensional input, the convolution operation is done between an input array and a two-dimensional array of weights, called as a filter or kernel. The filter is smaller than the input which is then systematically applied, through dot product, to each overlapping patch of the input data. The output of this operation is referred to as a feature map. (Brownlee 2019) Several of these feature maps are combined to detect higher level features and ultimately allow the network to classify an entire image (LeCun et al. 1999).

2.6.1 Visual Geometry Group (VGG)

Simonyan & Zisserman (2014) from the VGG of Oxford University discovered that deeper networks outperform shallower networks when they tested on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset. They proposed a convolutional neural network with an input image of size $224 \times 224 \times 3$ is passed through a stack of convolutional layers, containing between 16 to 19 layers, depending on the configuration. The filter size used in all the convolutional layers is the smallest possible receptive field, 3×3 , to capture all directions. The convolutional stride is constant at 1 in order to preserve the spatial resolution post the convolution operation. The number of channels increases in the power of two, from 64 to 512, for every group of convolution layers. After each group of convolutional layers, max pooling of size 2×2 and stride 2 is applied. The convolutional layers are then finally passed to three fully-connected layers - with the first two having 4096 channels and the last one containing

1000 channels as the model was trained on ImageNet which holds 1000 labels. (Simonyan & Zisserman 2014)

2.6.2 Xception

Szegedy et al. (2014) suggested that neural networks can achieve higher performances by not only getting deeper but also wider. This was achieved through a proposed Inception module where convolutional layers run parallel instead of stacked on top of each other as applied in VGG networks. The Xception architecture is built upon the Inception module therefore a brief overview is presented.

An Inception module calculates different transformations over the same input. This can be done by feeding the input to several convolutional layer sizes ranging from 1×1 , 3×3 and 5×5 . The outputs from these transformations are then combine through concatenation. Convolutional operations however are computationally expensive because it involves both spatial and depth-wise calculations. Therefore to reduce the depth of an input, a 1×1 convolution filter is applied depth-wise before performing the larger 3×3 and 5×5 convolutions. (Szegedy et al. 2014) Figure 2.4 illustrates the Inception module.

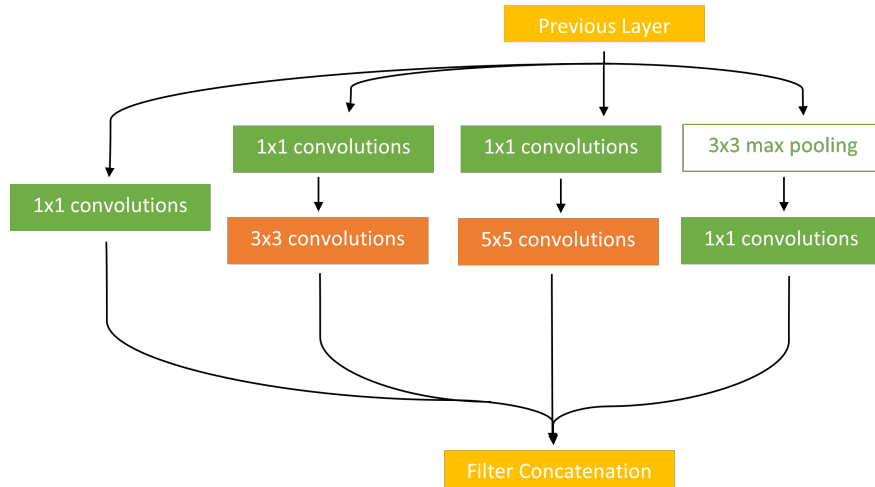


Figure 2.4: *Inception Module*

Inspired by the Inception module, Chollet (2016) put forward the Xception architecture which is fully based on depthwise separable convolutional layers. Through this type of convolution, the cross-channel and spatial correlations in a feature map are completely separated. In the Inception module, a 1×1 convolution is firstly applied to the input which will map the cross-channel correlations. The spatial 3×3 convolutions are then performed on each of these output channels. Finally, they are combined by concatenation as shown in Figure 2.5. The Inception architecture consists of 36 convolutional layers that form the feature extraction base, which are structured into 14 modules.

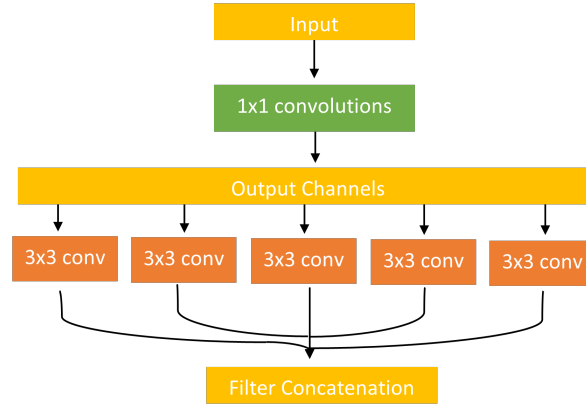


Figure 2.5: *Xception Module*

2.7 Summary

Current approaches to developing a multi-modal fake news classifier have leveraged Transformer-based models to extract text features and pre-trained deep CNNs to extract image features. Several of the works employed a simple concatenation of the two features before passing the combined features into a classification layer. However, there are several gaps identified which includes the lack of exploration of the choice of Transformer-based model, pre-trained CNNs, source task dataset, and the level of fine-tuning on the pre-trained models. Recognizing these gaps forms the basis of the experiments in this project.

An understanding of the main method of this project, transfer learning, was also developed to aid in the implementation of the model. Transfer learning can be used to extract features directly from a pre-trained model or several layers of a pre-trained model can be fine-tuned to the target task dataset. In addition, awareness of the differences in the popular Transformer-based models (BERT, RoBERTa, and XLNET) and deep CNNs (VGG and Xception) were laid out. The next chapter will detail the implementation of the multi-modal model and the proposed model architecture.

Chapter 3

Methodology

The main aim of this work is to design, implement, and evaluate a multi-modal fake news classifier on long contexts with one image. In the process of extracting the text features, several performances of pre-trained text transformers are compared. During the extraction of image features, performances of a few deep learning CNNs will be compared. Another experiment includes varying the source task dataset of the image model. Following the identification of the best performing image model, experimentation on the number of layers to retrain for fine-tuning will be conducted. Finally, the best performing text and image features will be combined through a dense layer model. This chapter presents the full methodology used in this project.

3.1 Dataset

The dataset used in this project comes from the FakeNewsNet repository (Shu et. al., 2019). FakeNewsNet is a comprehensive repository for multi-modal fake news research as it offers not only news content but also the social context, spatial and temporal information of the news piece. For the purposes of this project, the news content holding the text and image data will be used. The repository contains two datasets from different domains - PolitiFact for political news and Gossipcop for entertainment news. The PolitiFact dataset contains a total of 783 news articles whilst the Gossipcop dataset contains a total of 18,417 samples. Given the limited size of the PolitiFact dataset, this project only utilizes the Gossipcop dataset.

The news content database consists of 13 variables. This project will only utilize 3 variables: *title*, *text*, and *top img*. The variable *title* refers to the headline of the news, *text* contains the full news content, and *top img* provides the url of the image. A news piece may have multiple images however this project will only process the first image in the news piece as found in the *top img* variable.

3.1.1 Data Preparation

Before conducting the experiments, the raw dataset needs to be filtered to samples that contain at least one valid image. Similar to the approach of Singhal et. al. (2020), images that are either GIFs or icons were dropped. Further to their approach, images that are of insufficient size were also excluded. Size of the image was approximated by the Content-

Length of the url. The threshold applied was 5000 bytes. This was estimated by sampling several images in the database. All these filters were executed through a Python script using the Regular Expressions and Requests library. Lastly, logos and advertisements were manually removed as this information can increase the noise in the dataset. It is reasonable to assume that even reputable online media platforms contain advertising on their news content.

	Before	After
Real	16817	10120
Fake	5323	2906

Table 3.1: *Number of samples in the Gossipcop dataset before and after data preparation*

Table 3.1 clearly indicates that the dataset is highly imbalanced with over 77% of the samples in the real news class and only 22% in the fake news class. Methods that deal with an imbalanced dataset include undersampling and oversampling. However, these methods come with their own problems. Undersampling can lead to loss of information as samples from the majority class will have to be removed. As this project intends to apply deep learning models, working with a smaller dataset will not be the ideal choice. Oversampling, on the other hand, involves duplicating samples in the minority class which can lead to model overfitting and decreased model performance. Due to these issues, no change was made to the distribution of the target variable class. Class imbalance is addressed during the training process instead of being dealt with during the data pre-processing stage. It is essential to address the issue of imbalanced datasets because a model can be biased in predicting the majority class and become unable to learn enough features on the minority class (Cui, Jia, Lin, Song & Belongie 2019).

3.2 Proposed Model Architecture

The proposed model architecture was adapted from the multi-modal architecture of SpotFake+ (Singhal et al. 2020). A concatenation-based model was chosen over other types of models as majority of the work in multi-modal fake news classification employ this approach (Abdali 2022). Several simplifications were made such as reducing the number of dense layers and the number of nodes in each dense layers following unsatisfactory initial results on a more complex architecture.

Given a piece of news article, the features of both the text and visual components are extracted. These features are then passed through three fully connected layers of size 1024, 512, and 128, with a 40% drop out layer following the first fully connected layer. The 128-dimensional features of both text and images are then concatenated into a 256-dimensional feature which are then passed to another two fully connected layers of sizes 256 and 128. Finally, another 40% dropout layer is added before it goes through the classification layer. The activation function for the fully connected dense layers is relu whilst sigmoid was used for the classification layer. As there are only two classes, real or fake, the loss function used was binary cross-entropy. The chosen optimizer is Adam as used in Giachanou et al. (2020).

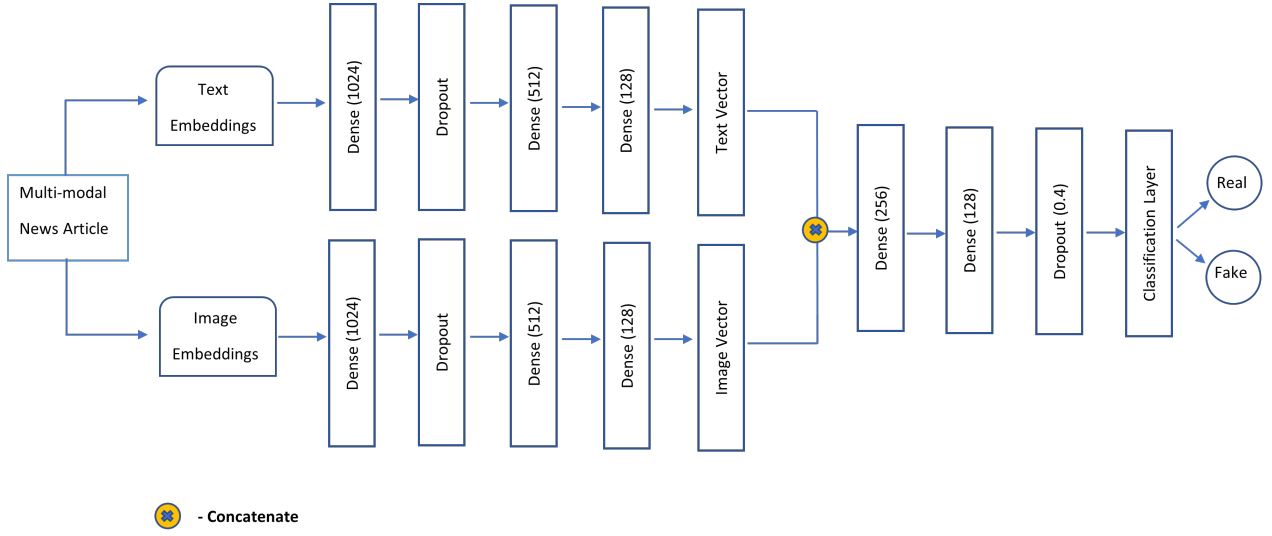


Figure 3.1: Proposed Architecture

3.3 Text Features

3.3.1 Text Pre-processing

Textual data usually contains a lot of noise and uninformative components that can negatively affect the performance of a text classifier. Given that this project utilizes several Transformer-based models, the text preprocessing applied is relatively light which largely aims to perform noise removal. This includes the removal of urls, punctuation and duplicate white spaces. Other preprocessing techniques such as stop word removal, stemming, and lemmatization were not performed as Transformer-based models are able to capture the contextual relationships between words. The HuggingFace Transformers library was used to perform the vectorization of the input strings.

3.3.2 Text Embeddings

Pre-trained encoder-based Transformer models can be used as feature extractors. The different Transformer models experimented on are BERT, RoBERTa, and XLNET as these are the most commonly used pre-trained models identified in the literature. Feature extraction can be achieved by loading a pre-trained model, freezing the weights of the model parameters, encoding the text input, and feeding it to the encoder stack. Each token from the text input returns a hidden state. According to the authors of BERT, the final hidden state corresponding to the classification token ([CLS]) of the last Transformer block sufficiently captures the entire context of the input sequence for a text classification task (Devlin et al. 2018). Hence, for all three pre-trained models, the text embeddings are derived from this representation.

For the BERT feature extraction, in addition to the text pre-processing steps mentioned above, the input were also transformed to lowercase as the original BERT base uncased model was chosen. No additional preprocessing steps were applied to the RoBERTa and XLNET embeddings as their based cased models were used. Through the *from_pretrained* method of the HuggingFace Transformers library, the weights of the three pre-trained models can be

loaded. Each pre-trained model has its own tokenizer function to encode the text input into a numerical representation. This can be performed through the *tokenizer.encode* method. As the pre-trained models expect inputs of uniform length, padding and truncation were applied, with the maximum sequence length set to 512. If the input sequence has more than 512 tokens, only the first 512 tokens will be taken. Whilst an input with less than 512 tokens will be filled with zero values to reach the maximum sequence length.

3.4 Image Features

Similar with the Transformer models, the pre-trained convolutional neural networks can be used for feature extraction. Through the Keras library, the pre-trained VGG16, VGG19, Xception, and VGGFace models can be loaded easily. When loading the models, the final dense layers are excluded. The inputs to the models are the raw images with sizes 224 x 224 and 299 x 299 for the VGG and Xception models, respectively. The raw images are then converted to numpy arrays. Each pre-trained model comes with its own preprocessing module through the *preprocess_input* function of the pre-trained model. The converted numpy arrays are passed to their respective *preprocess_input* function before calling the *predict* function in Keras which ultimately outputs the features from the pre-trained model. Two source datasets were used for their weights: ImageNet and VGGFace. There are limited network architectures available on the VGGFace model: VGG16, ResNet50, and SeNet50. To enable comparisons between the ImageNet and VGGFace models, only the VGG16 network weights were used on each pre-trained models.

For both VGG16 and VGG19, the image features were extracted from the last max pooling layer with dimension 7 x 7 x 512. For Xception, the features from the last global average pooling layer with a singular dimension of 2048 was used.

3.5 Data Splitting

In order to correctly merge the textual and visual features for the multi-modal classifier, the pre-processed dataset was split into their respective training, validation, and test sets during the feature extraction stage, while keeping the imbalance ratio constant across the sets. A random seed was fixed in order to keep the same samples for both modalities. The ratio of the split is 80-10-10, following the top performing results in Cui, Wang & Lee (2019) on the same dataset. An index variable is given to each sample as an identifier for later use during the merging process.

3.6 Weight Balancing During Training

In order to address the issue of training from an imbalanced dataset, a weighting factor is introduced to the loss function of the model. The weighting factor for each class is inversely proportional to their frequency. This weighting allows the model to give more weight to the minority class and can focus on reducing its error. The weight for each class can be computed as (Singh 2020):

$$w_i = \frac{n_{total}}{2 * n_i} \quad (3.1)$$

where n_{total} is the total number of samples in the dataset and n_i is the total number of samples in the given class. These weights were calculated based on the class frequencies of the training data which are then passed to the `class_weights` parameter in Keras within the fit model function. This weighting factor is ultimately multiplied to the loss function which means a weighted binary cross-entropy loss is used.

3.7 Metrics

The dataset is highly imbalanced with majority of the samples belonging in the real news category (target label - 0). Accuracy as a metric will be misleading as the model can simply classify all samples as real news and still achieve around 77% accuracy. To better account for the imbalanced nature of the data, the macro F_1 score is chosen as the primary metric. It is the arithmetic mean of the F_1 score for both classes. The formula for F_1 Score and its components are as follows:

$$F_1Score = 2 \frac{Precision * Recall}{Precision + Recall} \quad (3.2)$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (3.3)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (3.4)$$

where *True Positive* - measures the fraction of correctly classified fake news

True Negative - measures the fraction of correctly classified real news

False Positive - measures the fraction of incorrectly classified fake news

False Negative - measures the fraction of incorrectly classified real news

For comparison purposes with other multi-modal fake news models, accuracy, precision, and recall will also be reported. The *classification_report* function from scikit learn library was used to obtain the different metrics.

3.8 Summary

In this chapter, details of the dataset and its preparation before providing the inputs to the model were given. The proposed model architecture was described, along with its different components. The main components are the text and image features which are passed on several fully connected layers. More specific aspects of the model implementation and training were discussed such as the method for splitting the raw dataset and the weights applied to the loss function. Lastly, the metrics for evaluation were given. In the next chapter, the various experiments identified in Chapter 2 and its results as well as a final evaluation of the multi-model fake news classifier are reported.

Chapter 4

Experiments and Results

In this section, the details of the experimental design and results are reported. Each model under all the experiments were run 30 times to take into account the stochastic nature of the neural network models. The average values are subsequently reported.

4.1 Experiment 1: Text Embeddings

Design: The aim of this experiment is to identify the best performing pre-trained Transformer-based model as a feature extractor. These are BERT, RoBERTa, and XLNET. For all models, the extracted representation is the [CLS] token which is then passed through three dense layers with 1024, 512, and 128 nodes before going through the final classification layer. The activation and loss functions are similar to what was discussed in the previous section.

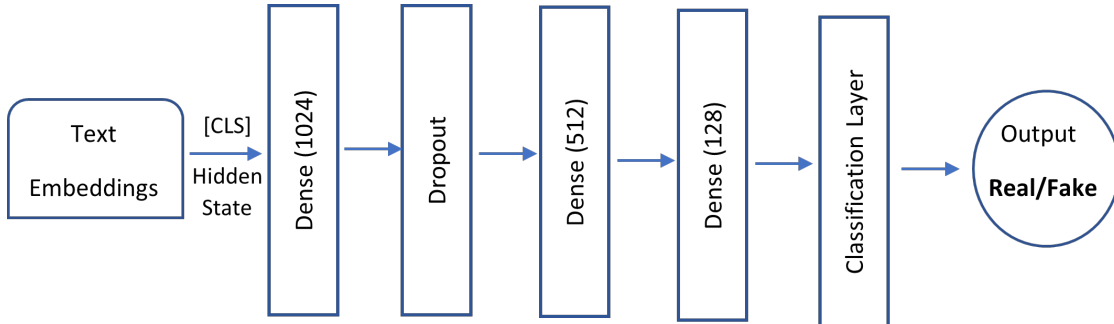


Figure 4.1: *Architecture of Text Only Model*

Process: Given that the proposed model architecture is a deep neural network, the model is prone to overfitting. Different methods to combat overfitting were used such as dropout, L2 regularization, early stopping, and a careful selection of the learning rate. As there are a number of possible combinations, the starting point was to refer to the methods applied in the SpotFake+ model (Singhal et al. 2020). In training the XLNET model in SpotFake+, an L2 regularization of 0.01 was applied on each dense layer, followed by a dropout rate of 0.4 before the classification layer, and trained on a learning rate of $1e^{-4}$. The other hyperparameters include training for 100 epochs and a batch size of 32. For all the three Transformer

Embedding	Macro-F1	SD of Macro-F1	Accuracy	Real			Fake		
				Precision	Recall	F1	Precision	Recall	F1
BERT	0.7281	0.00484	0.8056	0.8911	0.8562	0.8731	0.5487	0.6239	0.5831
RoBERTa	0.7457	0.00596	0.8199	0.8971	0.8696	0.8830	0.5800	0.6412	0.6085
XLNET	0.7501	0.00778	0.8288	0.8933	0.8877	0.8900	0.6128	0.6171	0.6101

Table 4.1: Comparing different textual embeddings

models, initially an L2 regularization of 0.01 was also applied to all the dense layers and without any additional overfitting methods included. However, this setting was still causing the models to overfit. The next method applied was early stopping to monitor the validation loss. A tolerance of three epochs was applied which means that the model will stop training if the validation loss does not improve after three epochs. This method worked very well however, it has caused the BERT and XLNET model to prematurely stop training which is evident in the training accuracy not yet plateauing. For the RoBERTa model, no further overfitting method was needed. The next method included was a dropout layer after the first dense layer, with values starting from 0.2 and increasing to 0.4. Lastly, in terms of learning rate, the BERT and RoBERTa models generalised better with a lower learning rate of $1e^{-5}$ compared with the XLNET model which worked best using a $1e^{-4}$ learning rate. All the models followed Singhal et al. (2020) for the number of training epochs and batch size. The progress of the training/validation accuracy and losses for each epoch and models can be found in the Appendix.

A summary of hyperparameters are as follows for each model:

Transformer Model	Learning Rate	L2 Regularisation	Dropout
BERT	1e-05	0.01 for all 3 dense layers	0.2 after 1st dense layer
RoBERTa	1e-05	0.01 for all 3 dense layers	None
XLNET	1e-04	0.01 for all 3 dense layers	0.4 after 1st dense layer

Result: Table 4.1 reports that the XLNET text embeddings achieves the highest macro-F1 score, outperforming BERT and RoBERTa by 2.2% and 0.44%, respectively. To statistically verify the primary evaluation metric, the macro-F1 score, an ANOVA test was conducted. The ANOVA test compares the means between three or more groups of equal sample sizes. Let μ_b be the mean macro F1-score of BERT, μ_r the mean macro F1-score of RoBERTa, and μ_x the mean macro F1-score of XLNET. The hypothesis can be expressed as:

$$H_0 : \mu_b = \mu_r = \mu_x \quad (4.1)$$

$$H_a : \mu_b \neq \mu_r \neq \mu_x \quad (4.2)$$

The p-value of the ANOVA test is considerably below 0.05 which is statistically significant at a 95% confidence level. A post-hoc test, Tukey's Honest Significant Difference (HSD) test, was also performed to determine which group of means are statistically significant. All possible pairs are tested. The p-value for all group comparisons are also below 0.05, which allows us to reject the null hypothesis and conclude that the means between all three groups are not equal.

Embedding	Macro-F1	SD of Macro-F1	Accuracy	Real			Fake		
				Precision	Recall	F1	Precision	Recall	F1
VGG16	0.5625	0.0105	0.7084	0.8088	0.8216	0.8148	0.3220	0.3026	0.3101
VGG19	0.5608	0.0148	0.7157	0.8073	0.8364	0.8212	0.3263	0.2827	0.3003
Xception	0.5947	0.00746	0.6904	0.8363	0.7512	0.7911	0.3473	0.4712	0.3984

Table 4.2: Comparing different image embeddings

4.2 Experiment 2 : Image Embeddings Without Fine Tuning

Design: The aim of this experiment is to identify the best performing pre-trained convolutional neural network model as a feature extractor. The models are VGG16, VGG19, and Xception. The architecture employed is similar to the first experiment where the features were passed through three dense layers with 1024, 512, and 128 nodes before it is fed to the final classification layer. For the VGG models, the features were first flattened before they are passed to the dense layers. The activation and loss functions are the same as those used in the first experiment.

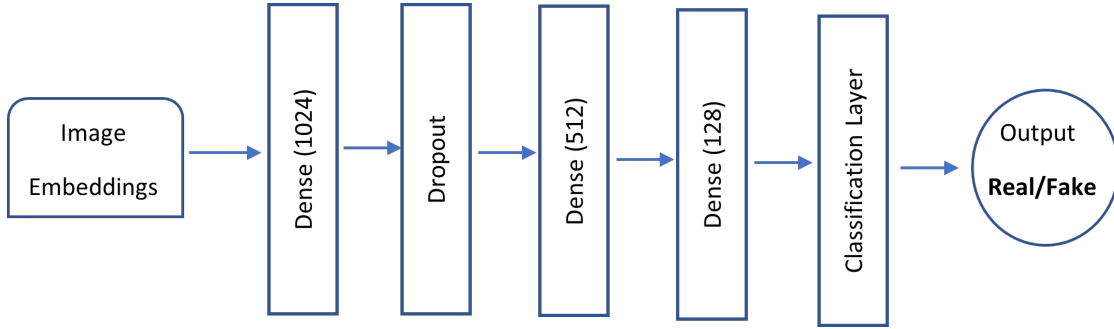


Figure 4.2: Architecture of Image Only Model

Process: Overfitting was also a concern in this experiment. As such, the same methodical approach as in the first experiment to combat overfitting were followed. During the training process, it was notable that the Xception model was quicker to overfit. Hence a lower learning rate of $1e^{-05}$ was used for Xception compared with $1e^{-04}$ for the VGG models. The progress of the training/validation accuracy and losses through each epoch and models can also be found in the Appendix.

A summary of hyperparameters are as follows for each model:

Transformer Model	Learning Rate	L2 Regularisation	Dropout
VGG16	1e-04	0.05 for the 1st dense layer, 0.1 for all other dense layers	0.2 after the 1st dense layer
VGG19	1e-04	0.05 for the 1st dense layer, 0.1 for all other dense layers	0.2 after the 1st dense layer
Xception	1e-05	0.01 for all 3 dense layers	0.4 after 1st dense layer

Result: Table 4.2 reports that the Xception architecture outperforms both the VGG architectures by 3% in terms of the macro-F1 score. The VGG19 architecture had the lowest macro-F1 score but outperforms the other architectures in terms of accuracy. This outper-

Embedding	Macro-F1	SD of Macro-F1	Accuracy	Real			Fake		
				Precision	Recall	F1	Precision	Recall	F1
VGG16	0.5625	0.0105	0.7084	0.8088	0.8216	0.8148	0.3220	0.3026	0.3101
VGGFace	0.5504	0.0114	0.6966	0.8044	0.8089	0.8059	0.3040	0.2938	0.2950

Table 4.3: Comparing different source dataset for image embeddings

formance is reflected on a higher F1 score for the negative class but not on the positive (fake news) class. For the ANOVA test on this experiment, let μ_{v1} be the mean macro F1-score of VGG16, μ_{v2} the mean macro F1-score of VGG19, and μ_{xc} the mean macro F1-score of Xception. The hypothesis can be expressed as:

$$H_0 : \mu_{v1} = \mu_{v2} = \mu_{xc} \quad (4.3)$$

$$H_a : \mu_{v1} \neq \mu_{v2} \neq \mu_{xc} \quad (4.4)$$

The p-value of the ANOVA test was below 0.05. However, the Tukey HSD test returned high p-values of 0.846 when comparing the means of VGG16 and VGG19. Therefore the null hypothesis cannot be rejected. However, the p-values when comparing the means between VGG16 and Xception as well as between VGG19 and Xception are below 0.05, indicating statistical significance at a 95% confidence level.

4.3 Experiment 3 : Image Embeddings From Different Source Task Datasets

Design: The aim of this experiment is to assess whether changing the dataset source of the pre-trained model can improve the performance of the image model. The same architecture as applied in Experiment 2 was used in this experiment. No changes were made to the hyperparameters.

Process: After the extracting the features from the pre-trained VGGFace model, the features were passed on to the same architecture as seen in Figure 4.2. The dimensions of the features are the same as the VGG16 features to have comparable results.

Result: Changing the dataset of the source task to VGGFace from ImageNet did not improve the performance of the image only model. The macro F1-score fell by 1.2%. A t-test was performed to determine whether there is statistical significance in the means of the two image embeddings. The hypothesis can be expressed as:

$$H_0 : \mu_{v1} = \mu_{v2} \quad (4.5)$$

$$H_a : \mu_{v1} \neq \mu_{v2} \quad (4.6)$$

The p-value of the t-test was less than the 0.05 therefore the null hypothesis can be rejected and conclude that there is a statistical significance between the two source task datasets.

4.4 Experiment 4: Effect of Fine-Tuning on the Best Performing Image Model

Design: The aim of this experiment is to assess whether fine-tuning the best performing image model so far, the Xception network, can improve performance. All the image features

extracted in the previous experiments did not include any fine-tuning. The choice of fine-tuning levels was based on the assigned number blocks and their respective layers in each flow. The entry flow is comprised of 4 blocks, the middle flow has 8 blocks, and the exit flow has 2 blocks. The control setting is the Xception model in Experiment 2 without any fine-tuning. For the first setting, all the layers in the exit flow are fine-tuned. In the second setting, all the layers in the exit flow plus all the layers in the last block of the middle flow are fine-tuned. As some initial results appear to have very minimal change on the performance, in the last setting the number of layers fine-tuned were increased up to all the layers in the last three blocks of the middle flow. The activation and loss function as well as the optimizer are unchanged from the settings of the previous experiments.

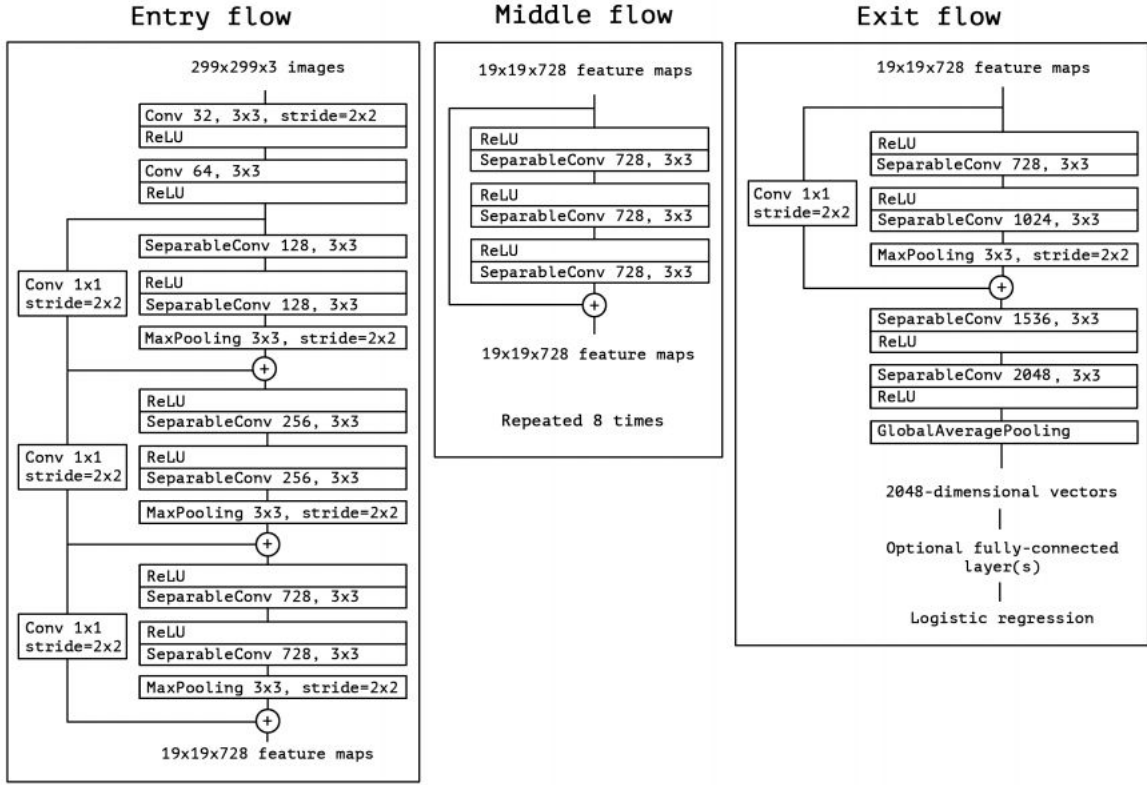


Figure 4.3: The Xception Architecture (Chollet 2016)

Process: Fine-tuning a pre-trained model involves freezing parts of the model and training the other parts of the model on the target dataset. The first step is to load the pre-trained Xception model through Keras with weights learned from ImageNet and the option to include the fully-connected layers is set to *False*. This is the base Xception model where all the layers are frozen. The loaded model outputs the last separable convolutional layer with dimensions $10 \times 10 \times 2048$. To be able to fine-tune the model, a global spatial average pooling layer, a fully connected layer with 2048 nodes, and a final classification layer (sigmoid) were added to follow the Xception architecture. In Experiment 2, the Xception network was found to overfit quickly therefore an L2 regularization of 0.1 was included in the fully connected layer. In addition, a 40% drop out layer followed the fully connected layer. For the purposes of the

Embedding	Macro-F1	SD of Macro-F1	Accuracy	Real			Fake		
				Precision	Recall	F1	Precision	Recall	F1
No Finetuning	0.5947	0.00746	0.6904	0.8363	0.7512	0.7911	0.3473	0.4712	0.3984
Last 17 Layers	0.5979	0.00523	0.7067	0.8313	0.7842	0.8068	0.3576	0.4288	0.3890
Last 27 Layers	0.6020	0.00337	0.7115	0.8326	0.7901	0.8106	0.3636	0.4298	0.3934
Last 57 Layers	0.5894	0.00572	0.6942	0.8295	0.7667	0.7967	0.3422	0.4344	0.3822

Table 4.4: Comparing different amounts of fine tuning

discussion, this is referred to as the new Xception model.

During the training process, the new Xception model will firstly be trained for 5 epochs at a higher learning rate of $1e^{-4}$. This is done to ensure that the base model with frozen layers and the added fully connected layers are trained to convergence. This is to avoid mixing randomly-initialised trainable layers with trainable layers containing pre-trained weights as such scenario can cause large gradient updates that can wipe out the pre-trained weights (Chollet 2020). After training for 5 epochs, the intended fine-tuned layers for each setting are then unfrozen. Next, the new Xception model will be once again trained as whole but at a lower learning rate of $1e^{-5}$ as overfitting is highly likely. This was done for 10 epochs with an early stopping strategy to monitor the validation loss with a tolerance of 3 epochs. Another training strategy adapted was to reduce the learning rate by a factor of 0.6 if validation loss still does not diminish after 3 epochs. Training and validation loss was monitored throughout the training process to ensure losses were converging. Refer to the Appendix section for these graphs. It was notable that as more layers were fine-tuned, it was faster for the model to overfit.

Once the training process was completed, the fine-tuned Xception model was saved and loaded to be used as a feature extractor as applied in Experiments 2 and 3. The extracted features are then applied to the same model architecture as displayed in Figure 4.2. The hyperparameters to all the fine-tuned models are the same as the Xception model in Experiment 2.

Result: Fine tuning had some success in improving the image only model. The best performance came from the second setting, tuning up to the last block of the middle flow for a total of 27 layers, which achieved a macro-F1 score of 60.20%. Table 4.4 reports that fine-tuning the Xception network improves the model performance after several layers but at a certain point it harms the model performance. In the third setting, the macro-F1 score fell by 1.26% compared with the second setting.

Similar to Experiments 1 and 2, an ANOVA test was performed to establish the statistical significance of the macro-F1 scores. For this experiment, let μ_{s0} , μ_{s1} , μ_{s2} , and μ_{s3} be the mean macro F1-score of no fine tuning, the first, second and third settings, respectively. The hypothesis can be expressed as:

$$H_0 : \mu_{s0} = \mu_{s1} = \mu_{s2} = \mu_{s3} \quad (4.7)$$

$$H_a : \mu_{s0} \neq \mu_{s1} \neq \mu_{s2} \neq \mu_{s3} \quad (4.8)$$

The p-value of the ANOVA test was below 0.05 therefore the null hypothesis can be rejected. However, the Tukey HSD test returned a high p-value of 0.143 when comparing the

Table 4.5: *Evaluation of Multi-Modal Classifier*

Macro F1	Accuracy	Real News			Fake News		
		Precision	Recall	F1	Precision	Recall	F1
0.6828	0.7467	0.8964	0.7645	0.8252	0.4470	0.6831	0.5404

means of the control setting and the first setting. Other pairs of means however were statistically significant at a the 95% confidence level. Therefore sufficient fine-tuning can make an improvement on pre-trained models but transferring too much can also harm the model.

4.5 Experiment 5: Combining the Best Performing Text and Image Models

Design: In this final experiment, the best performing text and image embeddings from Experiments 1 and 4 will be fused together to evaluate whether leveraging images can be beneficial. The model architecture applied is illustrated in Figure 3.1. The L2 regularization on the fully connected layers and the drop out hyperparameters for both the text and image streams of the model were based on the previous experiments. For the hyperparameters of the fully connected layers after the fusion of the two inputs, these were configured in this experiment. To evaluate the multi-modal classifier, in addition to the metrics reported in the previous experiments, a classification matrix will also be included in order to fully understand how well the model classifies both classes.

Process: The XLNET embeddings and the fine tuned Xception embeddings are loaded and merged by their assigned index values across the training, validation, and test sets. The most suitable framework to combine different inputs into a neural network architecture is through the use of the functional API of Keras. To configure whether the final two fully connected dense layers require any overfitting measures, the model was first run without any L2 regularization nor dropout layers. This configuration started to overfit by the 40th epoch when monitoring the training and validation loss. Next, a 0.001 L2 regularization was included on both of the final two fully connected dense layers. However, there is still some evidence of overfitting. Consequently, a dropout layer after the last dense layer was added, with a value starting at 0.1 and increasing to 0.4. The final model hyperparameters had 0.001 L2 regularization on both of the dense layers and a dropout of 0.4 after the last dense layer. The model had a learning rate of $1e^{-5}$ and was trained for 100 epochs but had an early stopping measure, with the same tolerance and metric as in the other experiments, in place. After training the model, it was evaluated on the test set on the number of True Positives, True Negatives, False Positives, and False Negatives. These metrics allowed the calculation of the precision, recall, and F1 scores of each class, as well as the production of a confusion matrix.

Result: The results of the model evaluation is reported in Table 4.5. The multi-modal achieved a macro-F1 score of 68.3% which is lower than the performance of the XLNET only model but higher than the best Xception model. This indicates that the image features has a negative attribution. The model is able to classify sufficiently the real news class but it is not successful with the fake news class which is of higher interest for this model.

The confusion matrix also verifies that the multi-modal model is unable to classify fake news given by the higher number of False Positive cases compared to the True Positive cases. There is a good number of True Negative cases but this is not the most important metric for this model.

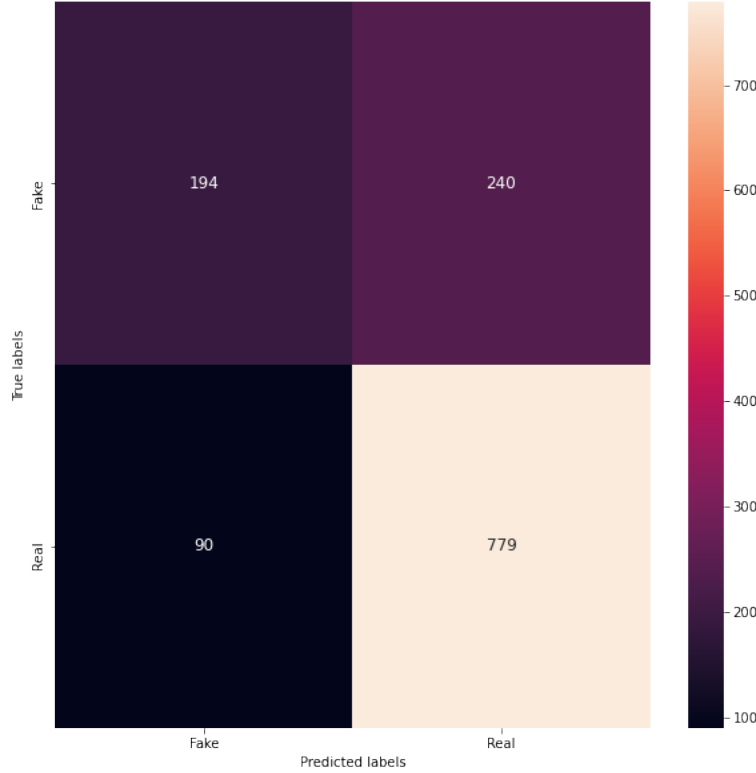


Figure 4.4: *Confusion Matrix of the Multi-Modal Model*

4.6 Summary

A total of five experiments were conducted in this project. The first experiment sought to identify the best performing Transformer-based model as a text feature extractor. The experiment identified that XLNET achieved a higher macro-F1 score than BERT and RoBERTa. In the second experiment, the top performing pre-trained CNN as an image feature extractor was established, with the Xception network outperforming both VGG networks. In the third experiment, it was established that changing the source task dataset from ImageNet to VGGFace did not yield better results. In the fourth experiment, it was imparted that fine-tuning has an impact on the model's performance but only up to a certain level. The last experiment involved combining the best performing text and image features for the proposed multi-modal fake news classifier. Evaluation of the multi-modal model was also presented. In the next chapter, a discussion and reflection upon these results are provided.

Chapter 5

Discussion

5.1 Experiment 1: Textual Embeddings

The results from Table 4.1 show that the choice of pre-trained Transformer model for extracting textual features do matter when building a fake news classifier. The improvements made on RoBERTa and XLNET over a few gaps identified in the BERT pre-training process was reflected by the presence of statistically significant differences between their means for this particular downstream classification task. Moreover, XLNET has a different architecture compared to both BERT and RoBERTa which suggests that permutation language modelling in XLNET has some advantage over masked language modeling that is applied in BERT and RoBERTa for capturing dependencies between tokens in a given sequence.

In terms of performances, BERT achieved a macro-F1 score of 72.81% which is below the reported 76.28% F1 score in Giachanou et al. (2020). A major difference in their BERT model implementation is that they only utilised the title of the article whereas in this project, the title and content of the article were included. The increased sequence length and subsequently the complexity of the input could have lead to a decrease in performance. This observation suggests that the model architecture in this project was insufficient to process the increased input complexity. For the best performing text embedding model, XLNET achieved an accuracy score of 82.9% which is only slightly below the accuracy reported in 83.6% the SpotFake+ model (Singhal et al. 2020). However, considering that the primary metric, macro-F1 score, only achieved 61% with XLNET on the fake news class, there is further room for improvement.

In building the text only model, several choices such as the Transformer representation and truncation techniques can be further experimented on to understand if other configurations can affect the performance. The representation of the [CLS] token was chosen in this project following similar multi-modal studies however, there are other possible representations to choose. In the work of Devlin et al. (2018) for a named entity recognition task, they utilized other representations such as summing the last four hidden layers and concatenating the last four hidden layers, where the latter achieved the best results. In terms of the truncation technique, the representations took the first 512 tokens of the title+content and discarded those after. This was an arbitrary choice therefore further experiments can be done to see if other truncation strategies work better such as taking the middle or the end of the text or splitting a given text and creating additional samples from it. Lastly, an interesting study

could be undertaken to understand whether analysing titles only or titles and the content of the article is more accurate in identifying fake news.

5.2 Experiment 2: Image Embeddings Without Fine-tuning

The results from Table 4.2 show that there is no difference in the features extracted from the 16 layer VGG network and the 19 layer VGG network when evaluated on a downstream classification task. Therefore, the intuition that a deeper network is better does not always hold true. There is however a statistical difference between the means of the VGG architectures and the Xception architecture. The Xception network has depthwise separable convolution layers rather than stacked convolution layers which allows this network to become more flexible in extracting hidden features for this particular dataset. The literature has indeed shown that the Xception architecture can outperform VGG networks (Raj & Meel 2021, Pashine et al. 2021). As new deep learning architectures are proposed, the multi-modal fake news research field also need to consider the latest architectures as the choice of networks for extracting features do matter.

As a control setting, there was an attempt to reproduce the image model presented in the SpotFake+ model where the extracted features from a VGG19 network are passed on to a larger and more fully connected dense layers. I also reached out to the authors to obtain the cleaned dataset they utilized for their experiments but I was unsuccessful with this. Reproducibility of the SpotFake+ results was not successful given that their results and methods were only presented in a 2 pager poster paper. They reported an accuracy score of 80% but after following their limited published methods, I was only able to achieve 71.7%, additionally they did not publish a macro-F1 score. Due to the difficulty in reproducing their results, this project used a smaller network which was able to achieve better results of 56.1% macro-F1 score versus 55% using their proposed architecture.

Feature extraction from a pre-trained model is the simplest way to leverage transfer learning. The results from this experiment show that this approach is insufficient as the best model, the Xception network, achieved a macro-F1 score below 50% on the minority fake news class. Therefore additional experiments were conducted to attempt to improve the performance.

5.3 Experiment 3: Image Embeddings From Different Source Task Datasets

The results of this experiment demonstrate that the choice of source task dataset has an effect on a fake news image classifier model. The VGGFace model was chosen based on the notion that the dataset trained on this model contains more similarity with the target task dataset at a higher level. In contrast to the ImageNet dataset which contains images of objects, the VGGFace dataset is composed of images of people which is also predominantly present in the Gossipcop dataset. The goal of transfer learning is to extract features from the source task and data and apply it to another task and data. The VGGFace model was not an improvement to the VGG16 model pre-trained on ImageNet perhaps because the Gossipcop dataset contained a larger amount of features than expected that do not characterize people

such as the background of an image or even stylized text embedded on the image. In selecting the most appropriate pre-trained model dataset, it would be beneficial to first measure the similarity between the source and target datasets.

Another drawback of the VGGFace dataset is its size compared to the ImageNet dataset, having 2.6M images versus 14M images for the latter. Therefore a model pre-trained on a larger dataset may be a better model as a feature extractor. It is not intuitive but the work of Neyshabur et al. (2020) evidenced that transfer learning does extract more low-level features from its source dataset than the higher level features. Therefore ImageNet is such a powerful dataset to leverage in computer vision because of the variety in the dataset as it contains over 20,000 different labels. The variety in the dataset then translates to transfer models being able to leverage more of the low-level features to another task or dataset.

Through an experimental approach, Bhattacharjee et al. (2020) concluded that both the size of the source dataset and the divergence of the source and task dataset are important factors in determining the success of transfer learning for the image classification task. There are several ways to measure the similarity between datasets such as the Kullback-Leibler Jensen-Shannon divergence measure or the Chi-square distance. In future works, it would be interesting to pre-train the neural network model on a Fake News multi-modal dataset such as NewsBag++, which contains 589K news articles with text and images, and then apply the learned features on a smaller dataset within the fake news space.

5.4 Experiment 4: Effect of Fine-Tuning On The Best Image Model

The presence of statistically significant results in this experiment suggests that finding the appropriate degree of fine-tuning when applying transfer learning has a measurable impact on the performance of a deep learning model. The level of fine-tuning must be treated as a hyperparameter in a model, similar to how learning rates and amount of regularization are selected in a model. This practice has not yet been seen in the fake news research field but has had some attention in the medical field (Lee et al. 2020). Another consideration when building models using transfer learning is the preference between accuracy and speed. Fine-tuning more layers equates to a greater number of parameters to be learned therefore a longer training time.

When retraining the layers closer to the classification layer on a smaller dataset, the target dataset's specific features are being learned. It is possible that a model's performance declines if more layers are retrained than necessary due to less transferable generalized features from the source task dataset. It would be useful to understand what are the actual features being transferred in each layer during the transfer learning process. Unfortunately, explainability in deep learning models remains a challenge.

5.5 Experiment 5: Multi-modal Fake News Classifier

Leveraging image features was not useful which is evidenced by a drop in the Macro-F1 score from the text only model of 75% to 68.3% for the multi-modal model. This is due to a lower performing image only model which only achieved a macro-F1 of 60.2%. The performance

of this multi-modal model however, outperformed the replicated SpotFake+ results in the work of Al Obaid et al. (2022) where the F1 score achieved was 49.76%. It is possible that the Gossipcop image data was homogeneous in nature and does not lend itself to being able to extract distinguishable predictive features. This finding is line with the work of Al Obaid et al. (2022) where they concluded that the usefulness of images in a multi-modal fake news model is dependent on the dataset. Giachanou et al. (2020) had a different conclusion where incorporating an image does achieve a higher F1 score compared to a text only model on the same dataset albeit very marginal at 2%.

It is difficult to establish a direct comparison across these models owing to different pre-processing techniques, model architecture, and model hyperparameters. However, it is clear that there are obvious gaps in the SpotFake+ model. The results are misleading when using accuracy as their primary metric given that the imbalanced nature of the Gossipcop dataset was not addressed. There was no confusion matrix to verify how well the model is able to classify the minority class, fake news. Singhal et al. (2019) also did not address the imbalanced nature of the Twitter and Weibo datasets but at a minimum they were able to report the precision, recall, and F1-score for each class. Therefore when comparing the results of different model architectures, awareness must be placed in analysing the different metrics and accounting for imbalance in the datasets.

Lastly, it is likely that a simple fusion of the text and image features through concatenation is ineffective given the inability of the validation accuracy to increase further after 20 epochs. This indicates that the model architecture is not sufficient as it is unable to generalize well to new data. Simple concatenation may have lead to a noisy representation of the two types of features. Perhaps incorporating an attention mechanism to both features can help the model find more important features for each modality.

5.6 Challenges in Multi-Modal Fake News Classifiers

This leads us to a larger issue when developing automated models that classify fake news. Is achieving the most accurate model a goal that will have the most impact in society? When deploying these models, should it be fully automated to filter out fake news completely from the media outlet or should there be an interface in which the end user can make the final judgement whether or not a piece of news article should be trusted or not? If the application leans towards the latter, then it is more beneficial to develop models that are able to provide explicable features that characterizes a piece of news as fake. End users will require interpretable and explainable models to build trust and confidence on these models.

Another main challenge is the dataset sources in which these models are trained on. Many multi-modal datasets learn event-specific news or domain-specific news such as those related to the US election or confined only to political or entertainment news (Yang et al. 2018, Shu et al. 2018). When models learn on domain-specific features, these models may not be very useful to new events. To explore this, an experiment could be proposed to compare the accuracy of some of these models on event specific datasets and more generalised fake news datasets. There is a need to leverage data that captures a wider set of news in order to develop more generalizable models. To achieve this, further work needs to be done to combine datasets from different domains and train models on a more varied dataset.

Chapter 6

Conclusions

Fake news matters as it has serious social, economic, and political impact with undesirable consequences like election interference and harm public health. This project designed, implemented, and evaluated a fake news classifier model that leverages two different types of input - text and images. In particular, it assessed how well transfer learning methods can extract features on both types of input for use in deep learning models. Deep learning models perform best with larger datasets. Transfer learning as the primary method was chosen due to the limited size of multi-modal datasets. The dataset utilized in this project was from Gossipcop which largely covers entertainment news. The features extracted were passed on to a series of fully connected dense layers, followed by concatenation of the two types of input, and further dense layers before the classification layer.

Several experiments were conducted in the process of building the classifier model. The first two experiments revealed that the choice of Transformer model and neural networks for extracting text and visual features have an impact on the model's performance. The outperformance of XLNET over BERT and RoBERTa suggests that permutation language modelling has an edge over masked language modelling as a source task whilst better results from the Xception network over either of the VGG networks suggests that depthwise separable convolutional layers has an advantage over stacks of convolutional networks in extracting generalizable image features. The third experiment imparted that the choice of source task dataset can also affect performance, with the popular ImageNet dataset more successful than the VGGFace dataset for this particular task. In the fourth experiment, the level of fine-tuning on the pre-trained models also had an impact on the model's result where fine-tuning improved the model's performance up to a certain point. As such selecting the degree of fine-tuning in transfer learning applications should be considered as another hyperparameter when designing a model.

The performance of the multi-modal model, leveraging XLNET and Xception networks to extract features, was unsatisfactory as it was unable to classify the minority class, the fake news class, owing to the image component of the model. Neither feature extraction nor fine-tuning using transfer learning for the image embeddings were sufficient for the Gossipcop dataset. Additional measures are necessary to help the deep learning models find distinguishable features between the two classes such as error level analysis to highlight tampered images as employed by Meel & Vishwakarma (2021) and attention mechanisms to focus on more significant parts of the features as found in the work of Sachan et al. (2021). There were

however good results from the text component of the model using simple transfer learning methods of feature extraction.

In future work, the main goal should be to develop more attributable and explainable machine learning methods as opposed to deep learning methods, as it is the opinion of this author that methods that are easier for the general public to understand the characteristics of fake news would have more value than developing the most accurate model. Additionally, as pointed out in the discussion section, there are many questions surrounding the most appropriate deep learning methods and choices that need to be answered.

Bibliography

- Abdali, S. (2022), ‘Multi-modal misinformation detection: Approaches, challenges and opportunities’, *arXiv preprint arXiv:2203.13883*.
- Al Obaid, A., Khotanlou, H., Mansoorizadeh, M. & Zabihzadeh, D. (2022), ‘Multimodal fake-news recognition using ensemble of deep learners’, *Entropy* **24**(9).
URL: <https://www.mdpi.com/1099-4300/24/9/1242>
- Ansar, W. & Goswami, S. (2021), ‘Combating the menace: A survey on characterization and detection of fake news from a data science perspective’, *International Journal of Information Management Data Insights* **1**(2), 100052.
- Bahdanau, D., Cho, K. & Bengio, Y. (2014), ‘Neural machine translation by jointly learning to align and translate’.
- Bhattacharjee, B., Kender, J. R., Hill, M., Dube, P., Huo, S., Glass, M. R., Belgodere, B., Pankanti, S., Codella, N. & Watson, P. (2020), P2l: Predicting transfer learning for images and semantic relations, in ‘Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops’.
- Brownlee, J. (2019), *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*, Machine Learning Mastery.
URL: <https://books.google.co.uk/books?id=DOamDwAAQBAJ>
- Chambers, S. (2020), ‘Truth, deliberative democracy, and the virtues of accuracy: Is fake news destroying the public sphere?’, *Political Studies* **69**(1), 147–163.
URL: <https://journals.sagepub.com/doi/full/10.1177/0032321719890811>
- Cho, K., van Merriënboer, B., Bahdanau, D. & Bengio, Y. (2014), ‘On the properties of neural machine translation: Encoder-decoder approaches’.
- Chollet, F. (2016), ‘Xception: Deep learning with depthwise separable convolutions’.
- Chollet, F. (2020), ‘Keras documentation: Transfer learning fine-tuning’.
URL: <https://keras.io/guides/transferlearning/>
- Cui, L., Wang, S. & Lee, D. (2019), Same: sentiment-aware multi-modal embedding for detecting fake news, in ‘2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)’, ASONAM ’19, ACM, NEW YORK, pp. 41–48.

- Cui, Y., Jia, M., Lin, T.-Y., Song, Y. & Belongie, S. (2019), Class-balanced loss based on effective number of samples, *in* ‘Proceedings of the IEEE/CVF conference on computer vision and pattern recognition’, pp. 9268–9277.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009), Imagenet: A large-scale hierarchical image database, *in* ‘CVPR09’.
- Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018), ‘Bert: Pre-training of deep bidirectional transformers for language understanding’.
- Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P. & Bengio, S. (2010), ‘Why does unsupervised pre-training help deep learning?’, *Journal of machine learning research* **11**, 625–660.
- Giachanou, A., Zhang, G. & Rosso, P. (2020), ‘Multimodal multi-image fake news detection’, *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)* .
- Giusti, S. & Piras, E. (2021), *In search of paradigms: Disinformation, fake news, and post-truth politics*, Routledge, p. 1–16.
- Goodfellow, I., Bengio, Y. & Courville, A. (2016), *Deep Learning*, MIT Press. <http://www.deeplearningbook.org>.
- Hangloo, S. & Arora, B. (2022), ‘Combating multimodal fake news on social media: methods, datasets, and future perspective’, *Multimedia Systems* .
- Howard, J. & Ruder, S. (2018), ‘Universal language model fine-tuning for text classification’.
- Jindal, S., Sood, R., Singh, R., Vatsa, M. & Chakraborty, T. (2020), Newsbag: A multimodal benchmark dataset for fake news detection, *in* ‘CEUR Workshop Proceedings’, Vol. 2560, pp. 138–145.
- Khan, J. Y., Khondaker, M. T. I., Afroz, S., Uddin, G. & Iqbal, A. (2021), ‘A benchmark study of machine learning models for online fake news detection’, *Machine Learning with Applications* **4**, 100032.
- LeCun, Y., Haffner, P., Bottou, L. & Bengio, Y. (1999), Object recognition with gradient-based learning, *in* ‘Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)’, Vol. 1681 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 319–345.
- Lee, K.-S., Kim, J. Y., Jeon, E.-t., Choi, W. S., Kim, N. H. & Lee, K. Y. (2020), ‘Evaluation of scalability and degree of fine-tuning of deep convolutional neural networks for covid-19 screening on chest x-ray images using explainable deep-learning algorithm’, *Journal of Personalized Medicine* **10**(4), 213.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. & Stoyanov, V. (2019), ‘Roberta: A robustly optimized bert pretraining approach’.

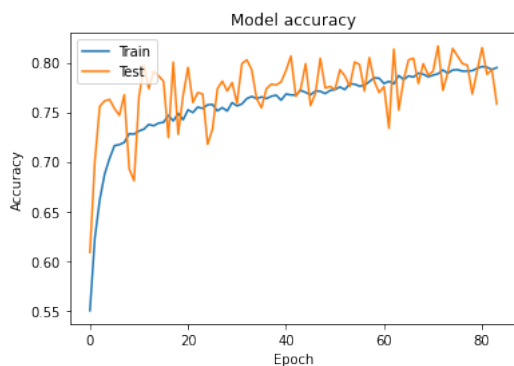
- Meel, P. & Vishwakarma, D. K. (2021), ‘Han, image captioning, and forensics ensemble multimodal fake news detection’, *Information Sciences* **567**, 23–41.
URL: <https://www.sciencedirect.com/science/article/pii/S0020025521002826>
- Mridha, M. F., Keya, A. J., Hamid, M. A., Monowar, M. M. & Rahman, M. S. (2021), ‘A comprehensive review on fake news detection with deep learning’, *IEEE access* **9**, 156151–156170.
- Neyshabur, B., Sedghi, H. & Zhang, C. (2020), ‘What is being transferred in transfer learning?’, *Advances in neural information processing systems* **33**, 512–523.
- Ofcom (2020), *News Consumption in the UK: 2020*.
URL: https://www.ofcom.org.uk/data/assets/pdf_file/0013/201316/news-consumption-2020-report.pdf
- Pan, S. J. & Yang, Q. (2010), ‘A survey on transfer learning’, *IEEE Transactions on Knowledge and Data Engineering* **22**(10), 1345–1359.
URL: <https://browzine.com/articles/4490456>
- Parkhi, O. M., Vedaldi, A. & Zisserman, A. (2015), Deep face recognition, in ‘British Machine Vision Conference’.
- Pashine, S., Mandiya, S., Gupta, P. & Sheikh, R. (2021), ‘Deep fake detection: Survey of facial manipulation detection solutions’.
- Pilehvar, M. T. & Camacho-Collados, J. (2020), *Embeddings in natural language processing : theory and advances in vector representations of meaning*, Morgan & Claypool Publishers.
- Plested, J. & Gedeon, T. (2022), ‘Deep transfer learning for image classification: a survey’.
- Raj, C. & Meel, P. (2021), ‘Convnet frameworks for multi-modal fake news detection’, *Applied Intelligence* **51**(11), 8132–8148.
- Ruder, S. (2019), Neural transfer learning for natural language processing, PhD thesis.
URL: https://ruder.io/thesis/neural_transfer_learning_for_nlp.pdf
- Sachan, T., Pinnaparaju, N., Gupta, M. & Varma, V. (2021), Scate: shared cross attention transformer encoders for multimodal fake news detection, in ‘Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining’, pp. 399–406.
- Shu, K., Mahudeswaran, D., Wang, S., Lee, D. & Liu, H. (2018), ‘Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media’, *arXiv preprint arXiv:1809.01286*.
- Simonyan, K. & Zisserman, A. (2014), ‘Very deep convolutional networks for large-scale image recognition’.
- Singh, K. (2020), ‘How to improve class imbalance using class weights in machine learning’.
URL: <https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/>

- Singhal, S., Kabra, A., Sharma, M., Shah, R. R., Chakraborty, T. & Kumaraguru, P. (2020), ‘Spotfake+: A multimodal framework for fake news detection via transfer learning (student abstract)’, *Proceedings of the AAAI Conference on Artificial Intelligence* **34**(10), 13915–13916.
URL: <https://ojs.aaai.org/index.php/AAAI/article/view/7230>
- Singhal, S., Shah, R. R., Chakraborty, T., Kumaraguru, P. & Satoh, S. (2019), Spotfake: A multi-modal framework for fake news detection, *in* ‘2019 IEEE fifth international conference on multimedia big data (BigMM)’, IEEE, pp. 39–47.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2014), ‘Going deeper with convolutions’.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017), ‘Attention is all you need’.
- Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z. & Yu, P. S. (2018), ‘Ti-cnn: Convolutional neural networks for fake news detection’.
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. & Le, Q. V. (2019), ‘Xlnet: Generalized autoregressive pretraining for language understanding’.

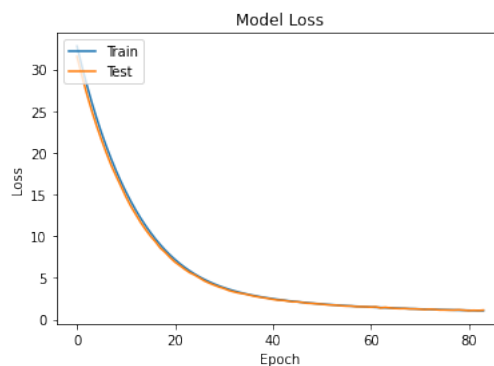
Appendices

Appendix A

Monitoring Training

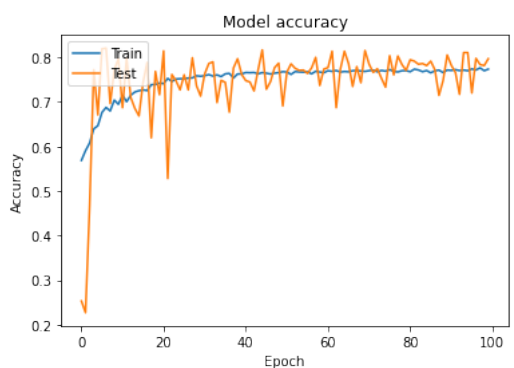


(a) Training and Validation Accuracy

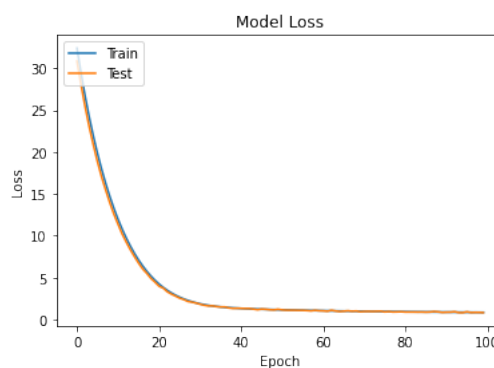


(b) Training and Validation Loss

Figure A.1: Monitoring Training on the BERT Model

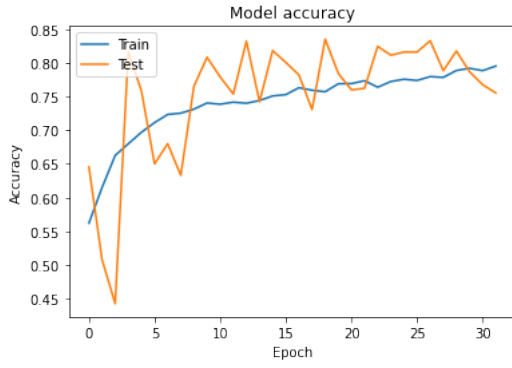


(a) Training and Validation Accuracy

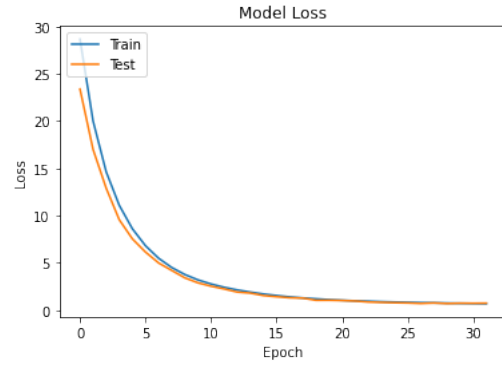


(b) Training and Validation Loss

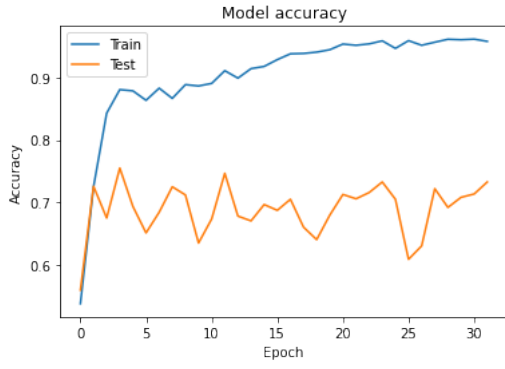
Figure A.2: Monitoring Training on the RoBERTa Model



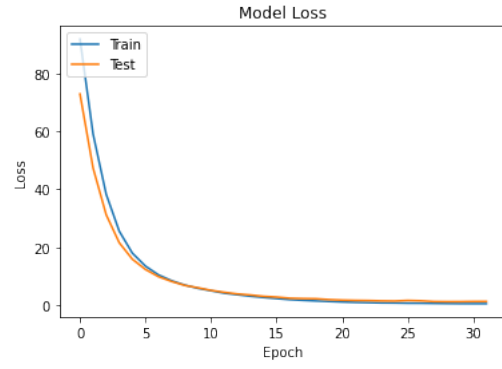
(a) Training and Validation Accuracy



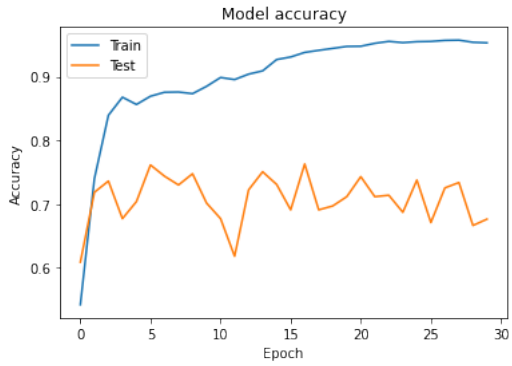
(b) Training and Validation Loss

Figure A.3: Monitoring Training on the XLNET Model

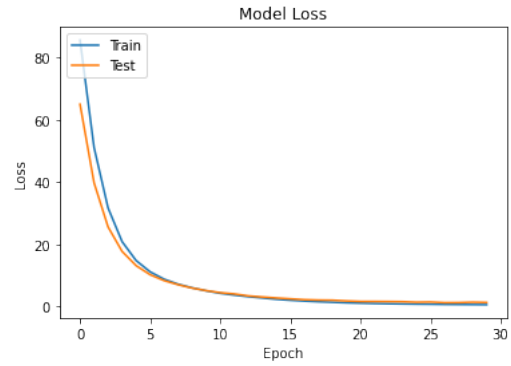
(a) Training and Validation Accuracy



(b) Training and Validation Loss

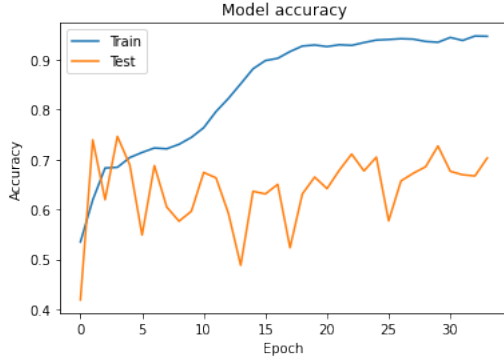
Figure A.4: Monitoring Training on the VGG16 Model

(a) Training and Validation Accuracy

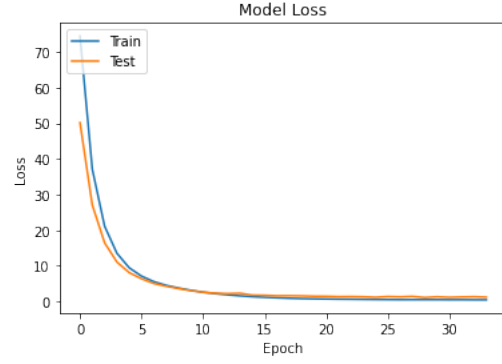


(b) Training and Validation Loss

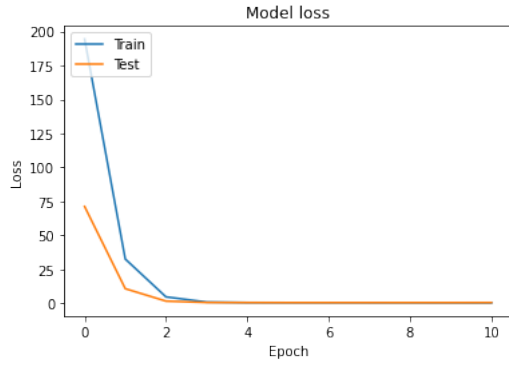
Figure A.5: Monitoring Training on the VGG19 Model**Figure A.6:** Monitoring Training on the Xception Without Fine Tuning Model



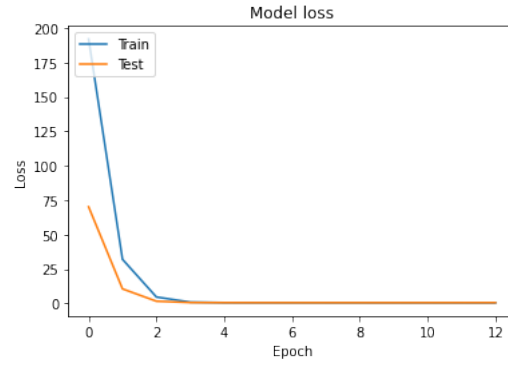
(a) Training and Validation Accuracy



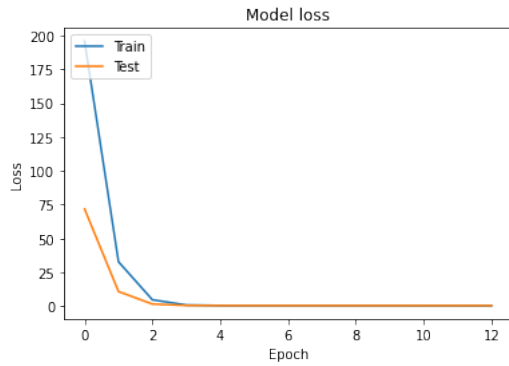
(b) Training and Validation Loss

Figure A.7: Monitoring Training on the VGGFace 16 Model

(a) Fine Tuning to Block 9

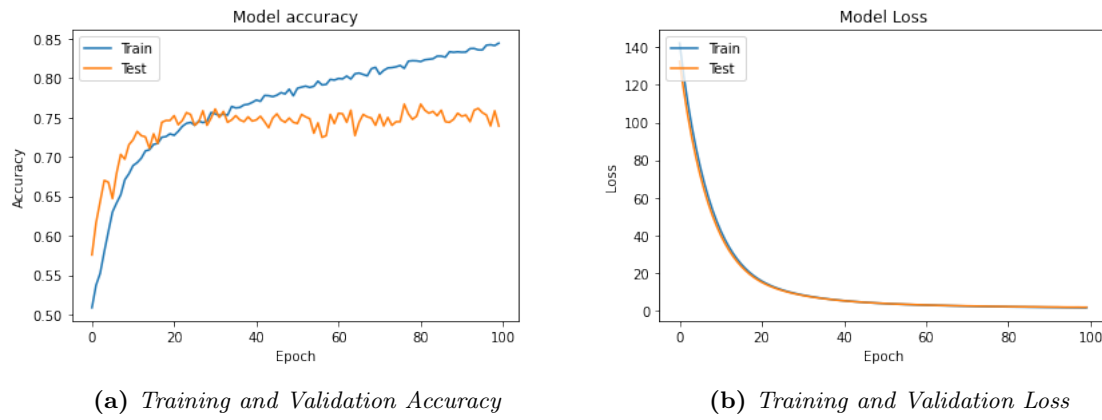


(b) Fine Tuning to Block 12



(c) Fine Tuning to Block 13

Figure A.8: Monitoring Training on the Different Levels of Fine Tuning on Xception Network

**Figure A.9:** Monitoring Training on the Multi-Modal Model