A Deep Learning Framework for Multi-Modal Fake News Detection

Evangelos Tsogkas
Department of Informatics
Athens University of Economics and Business
Athens, Greece
Email: vagelist4@gmail.com

Vana Kalogeraki
Department of Informatics
Athens University of Economics and Business
Athens, Greece
Email: vana@aueb.gr

***Abstract*—News consumption on social media has become extremely popular and attracts more and more people due to the easy access, timeliness and low cost. However, at the same time, the environment of social media proves to be convenient for rapid dissemination of fake news that aim to mislead readers providing deliberate disinformation. Fake news have severe negative effects on public trust, government, economy and journalism. Therefore, fake news detection is non-trivial, but existing propagation-based methods, although promising, are effective only after fake news have already spread. On the other hand, most content-based methods focus only on features extracted from text, ignoring the images that news published on social media often contain. In this work, we propose a deep learning framework for real-time fake news detection that combines both text and visual content. We also compare different word embeddings and state-of-the-art models for image feature extraction for effective learning of latent text and visual features. Through detailed experiments conducted on two real-world datasets, we show that taking advantage of visual content is capable of improving the performance of fake news detection compared to text content alone.**

I. Introduction

As social media platforms have grown considerably and many people spend an increasing amount of their time interacting through them, they have also become a primary source for receiving news and information. The Pew Research Center reported that in 2018, 68% of U.S adults get news from social media and this percentage has been continuously increasing from 49% in 2012 [1]. This increase of news consumption on social media has sparked the wide spread of fake news as they can reach more people, while at the same time it is easy and cheap to publish news online and faster to disseminate. A great example of fake news reach was the 2016 U.S presidential election campaign, where the top twenty most discussed fake election stories generated 8,711,000 shares, reactions and comments on Facebook, while the top twenty most discussed election stories posted by 19 major news websites generated less, with a total of 7,367,000 [2]. Fake news have many negative effects both on individuals and society. They can hurt the trust of people for journalism, manipulate their way of thinking, opinions and actions and can even affect the economy. For example, fake news claiming that Barack Obama was injured in an explosion wiped out $130 billion in stock value [2]. Thus, it is evident that there is need of a fake news detection system in social media.

Fake news detection is hard, as it is not clear exactly what distinguishes fake news from real news. Studies have shown that the human is not capable of performing much better than simple chance of guessing correctly when it comes to detecting deception. In fact, the mean accuracy over 1000 participants in more than 100 experiments is only 54% [2]. Fact-checking websites like PolitiFact[1] and FactCheck.org[2] have been created with the goal to alleviate this problem by providing the ground truth for latest news. In such websites, a group of professional fact-checkers examine all sides, claims and conclusions and by gathering knowledge they try to identify as accurately as possible whether a piece of news is fake or real. An example of labeling news from PolitFact's website is shown in Figure 1.

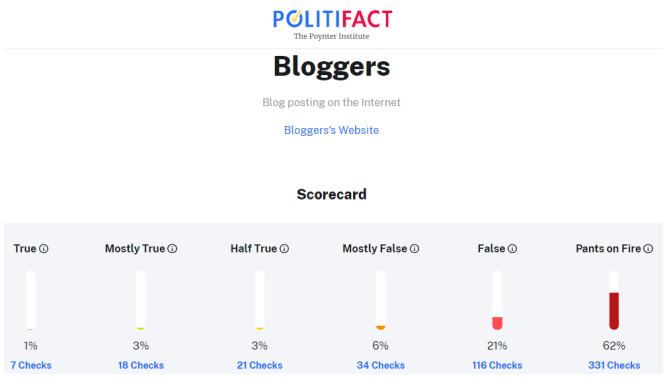


Fig. 1: Example of a PolitiFact's score card for online blog posts, which shows the percentages of statements labeled based on their truthfulness.

Manual fact-checking, however, is costly and extremely difficult to scale up considering the vast amount of news published online every day, so there is a lot of ongoing research on automatic fake news detection. Traditional machine learning approaches that try to extract features from news content [3] cannot provide high quality results, as fake news are often

[1] www.politifact.com
[2] www.factcheck.org

written in a way that is similar to real news for successful deception. Deep learning approaches [1] are more effective, because they can extract more meaningful features from text, but they are still lacking. In order to improve the performance of text-based fake news detection, a few works have effectively utilized features extracted from images. Such features are explored in this work [4], yet, they are still handcrafted and not competent enough to represent complex distributions of visual content, while deep learning based models for extracting features from images [5] have shown promising results.

Images are popular in news published online. By providing visual content, both short posts and full-length articles can not only catch the attention of readers, but they also offer a vivid content with better explainability, which makes news reading more pleasant and easier. In fact, the dissemination of news containing images is even faster. In the dataset proposed in this work [4], the average re-tweet number of news containing images is 11 times larger than that of news containing only text. Therefore, images can play an important role in news dissemination and publishers of fake news can take advantage of that.

Acknowledging the problem of fake news in social media and the importance of images in news content, in this work, we propose a deep learning framework that can extract features both from text and visual content and combine them for improved results in fake news detection. Our aim is to provide a model that can detect fake news in real-time as accurately as possible, without relying on late detection based on propagation and social context, by taking advantage not only of the text content, but the visual content, too. In order to learn more effective latent features from text and images, we also compared different word embeddings and state-of-the-art deep learning models for image feature extraction.

We employed a 1D CNN to extract features from text using pre-trained word embeddings, including Word2Vec [6], GloVe [7] and FastText [8]. For image feature extraction we made use of deep learning models with pre-trained weights on ImageNet[3], including VGG-19 [9], ResNet-152 V2 [10], MobileNet V2 [11], DenseNet-201 [12], Inception V3 [13], NasNet Large [14] and Xception [15]. Our experiments conducted on two real-world datasets show that combining text and visual features can lead to better results in detecting fake news compared to using only text features. We also found that the pre-trained FastText word embeddings produced better results in text feature extraction, while Xception produced better results in extracting features from images.

The main contributions of our work are summarized as follows:

1) We propose a model that can detect fake news in real-time by combining features extracted from both text and visual content.
2) We compare different word embeddings and state-of-the-art deep learning models for effective text and visual feature extraction.
3) We conduct extensive experiments on two real-world datasets that contain news articles of varying length, to demonstrate the effectiveness of our model for detecting fake news by combining text and visual content.

The rest of the paper is organized as follows. In section II we summarize related work on fake news detection. In Section III we introduce our model and the details of its architecture. In Section IV we describe the datasets and present the experimental results. Finally, we conclude in Section V.

II. Related Work

In recent years a lot of research has focused on issues related to fake news. For effective detection of fake news it is challenging to understand what makes them fake and how to distinguish them from real news. Xinyi et al. [2] gathered and presented fundamental theories regarding human cognition and behavior in order to study fake news from different perspectives, such as how they are written, how they spread and how users engage with them. For example, according to the *Validity effect* [16], individuals tend to believe information is correct after repeated exposures, which is a serious problem considering how fast fake news disseminate in social media.

Many works have focused on different aspects related to fake news, trying to extract useful features for their detection. Some researches have extensively studied language features related to fake news detection. For example, Pérez-Rosas et al. [17] propose features such as n-grams and CFGs based on TF-IDF encoding. Xinyi et al. [3] explore and compare various linguistic feature groups at lexicon, syntax and semantic level to detect fake news using traditional machine learning algorithms. Deep learning models have proved to be even more successful as they skip feature engineering and they can learn latent text features able to better represent the difference between fake and real news [18], [19].

Another approach for detecting fake news is the study of the propagation patterns and the social context. Wu et al. [20] propose a graph-kernel based hybrid SVM classifier to capture high-order propagation patterns in addition to semantic features such as sentiment and topics. Shu et al. [21] exploit user profile features to identify which users are more likely to share fake news and Shu et al. [22] create a tri-relationship embedding framework to model the relations between news, publishers and users. Exploiting propagation and social context has shown successful results in detecting fake news, however these approaches are only effective after fake news have already spread in social media.

In order to provide even more effective models for the detection of fake news some works have combined features from both text content and user comments [1] or even different modalities of news content, i.e. text and visual content. Many

[3]www.image-net.org

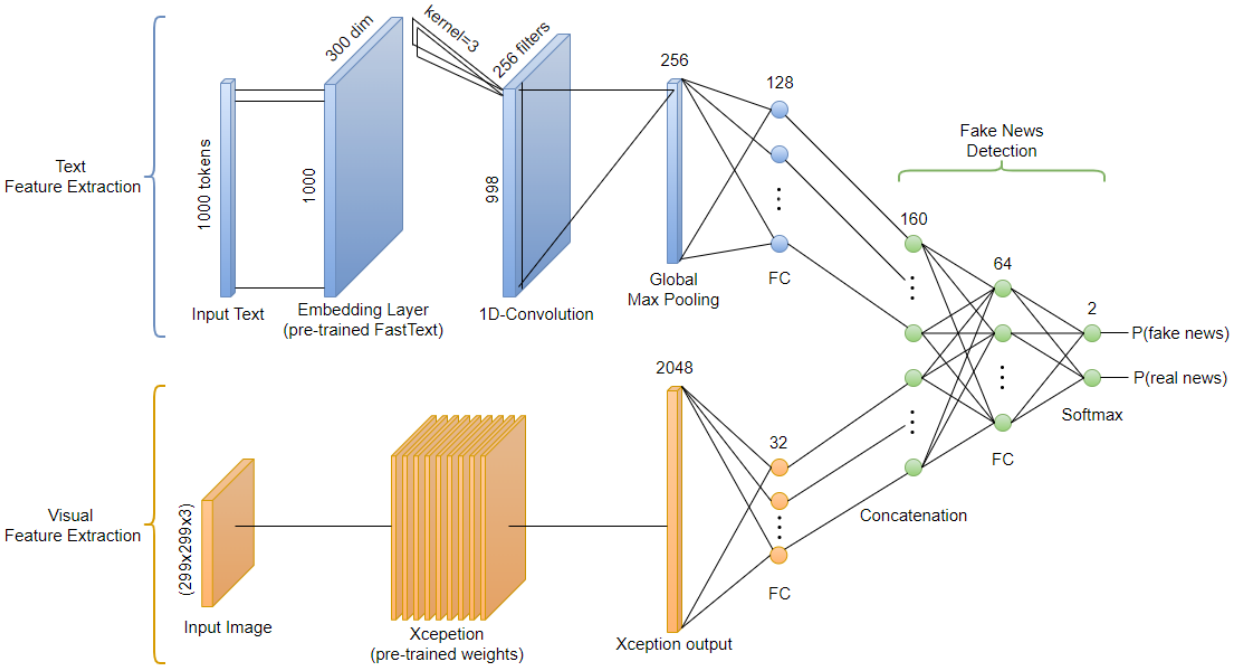


Fig. 2: The architecture of the proposed model.

features extracted from images have been studied in this work [4], both visual, such as the average of visual similarities between image pairs, and statistical, such as the ratio of most popular images. Such handcrafted features, however, are still not so effective at capturing complex visual distributions. Wang et al. [5] use a deep learning model for successful image feature extraction and also propose an event discriminator to learn event invariant features for better generalization.

In this paper we propose a deep learning framework that can combine text and visual content and we compare word embeddings and state-of-the-art deep learning models for image feature extraction to improve the features learned from both text and images. We also study the effect of combining such features on datasets that contain news articles of varying length.

III. Proposed Model

In this section, we present the details of the proposed model for real-time fake news detection combining text and visual content. As shown in Figure 2 the model consists of three major parts:

1) The text feature extraction.
2) The visual feature extraction.
3) The fake news detection by combining the text and visual features.

A. Text feature extraction

The input text is in the form of a sequence consisting of 1000 tokens. News articles can be of varying length, thus if the text contains less tokens the sequence is padded with zero values so that all text sequences have the same size. The tokens are then mapped to vectors of dimension 300 in the embedding layer. After experimenting with different word embeddings we selected FastText [8] pre-trained vectors for that purpose, in order to avoid over-fitting by training embeddings from scratch on small datasets. After the embedding layer we apply a small dropout rate of 0.2 as it helps in generalization. To extract text features we deploy a 1D convolution layer that has 256 filters and a kernel of size 3, i.e, it creates a feature vector of dimension 256 by parsing the embedding vectors of 3 words each time and the kernel then shifts by one word. The ReLU activation function is used for the convolution. After all feature vectors are created by the convolution process we use the global max-pooling operation to keep the most important information by taking the maximum values. This results in a vector of dimension 256. Finally, a fully connected layer with 128 hidden nodes is used and the ReLU activation function as well as a dropout regularization of 0.5 are applied to its output.

B. Visual feature extraction

We use the images attached by the news articles as input for the image feature extraction. A lot of deep learning models for effective image classification have been trained on the ImageNet dataset that contains millions of images. After extensive experiments we selected the Xception model [15] as we found that it is able to extract more effective features from images. The default input size for this model is 299x299 with exactly 3 channels. For the feature extraction we use the pre-trained weights of the model on ImageNet, without

updating them through the training process, in order to avoid over-fitting. The image features we get from Xception is the output of the global average-pooling operation before the final output layer. We then pass this vector of dimension 2048 through a fully connected layer with 32 hidden nodes, the ReLU activation function and a dropout layer with a rate of 0.5.

C. Fake news detection

In order to detect fake news we first merge the features extracted from text content (128 dimension) and those extracted from the images (32 dimension) by concatenating them, which results to a layer with 160 hidden nodes. After that, a fully connected layer follows with 64 hidden nodes, a ReLU activation and a dropout regularization of 0.5. Finally, we use the softmax function to get the output of the fake news detection in the form of probabilities. Specifically, the output consists of two values, the probability of the news article being fake and the probability of it being real. Depending on which is the highest probability it is finally labelled accordingly.

D. Implementation details

For the implementation of the model we use the Keras [23] high-level API of Tensorflow [24]. The model is built using the functional API of Keras to jointly train and concatenate the models for the text and image feature extraction and finally for the fake news detection. We minimize the categorical cross-entropy loss and to seek the optimal parameters we use the Adam optimizer [25] with learning rate 0.001. The instances come in batches of 32 and we adopt the early stopping strategy to avoid over-fitting, by setting aside a 10% validation set to monitor the validation loss. For the image feature extraction we use the Xception model available through the Keras applications with pre-trained weights on ImageNet.

IV. Experiments

In this section, we first describe the two datasets used in the experiments. We then present the experimental results of comparing different word embeddings and state-of-the-art deep learning models for image feature extraction. Finally, we analyze the performance of the proposed model that combines text and visual features.

A. Datasets

We make use of the *FakeNewsNet* [26], [27], a comprehensive fake news detection benchmark dataset, which consists of two individual datasets: *PolitiFact* and *GossipCop*, both containing labeled real-world news content and social context information. The datasets contain both short posts and full-length articles. For our experiments we only use the body text and attached image of the news content. Next, we introduce the details of these datasets whose statistics are shown in Table I.

TABLE I
Statistics of FakeNewsNet dataset

Platform	PolitiFact	GossipCop
# Fake News	231	3190
# Real News	278	3268
# Total News	509	6458

1) ***PolitiFact****:* The *PolitiFact* dataset is created by obtaining news from the PolitiFact fact-checking platform. In this platform, professional fact-checkers review news and evaluate their truthfulness as previously shown in Figure 1. These evaluations are used as the ground truth to label the dataset's news as fake or real. The news in the *PolitiFact* dataset are mostly related to politics, but it also contains some other topics, such as health and science.

2) ***GossipCop****:* GossipCop[4] is a website that fact-checks news stories related to entertainment and celebrities. Specifically, it provides rating scores on the scale of 0 to 10 which represent the degree of a news story being fake or real. Most of the news provided by this website have a rating lower than 5 and these are used as the fake news of the *GossipCop* dataset. The real news of the dataset, also related to entertainment and celebrities, are crawled from E! Online[5], a well-known trusted media website.

We crawled the news of both datasets using the script available in the *FakeNewsNet* GitHub repository[6]. The images attached by the news are not downloaded, but their URLs are available, so we downloaded them separately by creating another crawler. As our goal is to build a model that can detect fake news by combining text and visual content, we only kept news that contain both a body text and an attached image. We also removed news pieces whose images have a width or height smaller than 75, because of input size limitations of some deep learning models used for image feature extraction. We split both datasets into 80% training and 20% testing sets, by using the train_test_split method of scikit-learn [28] and the stratify parameter, in order to keep the same ratio of the two classes in both training and test set. The same splits are used for all experiments.

B. Text model performance

In this subsection, we provide the details of the pre-trained word embeddings used in the experiments and describe the models that produced them. We then show the experimental results of comparing their performance when used for the detection of fake news with text content alone.

1) **Word2Vec** [6]: We use embedding vectors pre-trained with the Word2Vec model on part of the Google News

[4]www.gossipcop.com
[5]www.eonline.com
[6]github.com/KaiDMML/FakeNewsNet

TABLE II
PERFORMANCE COMPARISON OF MODELS USING ONLY TEXT CONTENT

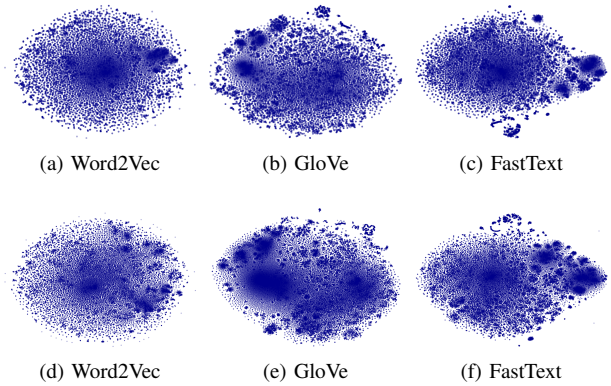
Dataset	Model	Accuracy	Fake News			Real News		
			Precision	Recall	$F1$	Precision	Recall	$F1$
PolitiFact	Logistic Regression	0.794	0.745	**0.826**	0.784	**0.843**	0.768	0.804
	CNN (Word2Vec)	0.794	0.791	0.739	0.764	0.797	0.839	0.817
	CNN (GloVe)	0.804	**0.795**	0.761	0.778	0.810	**0.839**	0.824
	CNN (FastText)	**0.814**	0.787	0.804	**0.796**	0.836	0.821	**0.829**
GossipCop	Logistic Regression	0.727	0.726	0.719	0.723	0.729	0.735	0.732
	CNN (Word2Vec)	0.745	0.778	0.676	0.723	0.720	0.812	0.763
	CNN (GloVe)	0.767	**0.793**	0.715	0.752	0.746	**0.818**	**0.780**
	CNN (FastText)	**0.769**	0.782	**0.738**	**0.759**	**0.758**	0.799	0.778

dataset consisting of about 100 billion words. The embeddings are 300-dimensional vectors for 3 million words and phrases. The Word2Vec model can utilize two architectures for creating these vector representations: the continuous bag-of-words and continuous skip-gram. Depending on the algorithm used it either predicts the central word from a window of surrounding context words(bag-of-words) or uses the central word to predict the other words of the window(skip-gram).

2) **GloVe** [7]: We use the pre-trained GloVe embeddings on 42 billion tokens of Common Crawl, which are 300-dimensional vectors for 1.9 million words. The GloVe (Global Vectors) model, to produce the vector representations of words it is trained on non-zero entries of a global word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus. This way, it also uses statistical information for the word representations.
3) **FastText** [8]: For FastText embeddings we use 2 million word vectors pre-trained on 600 billion tokens of Common Crawl. The FastText model is based on skip-gram, but each word is represented as a bag of character n-grams. By using sub-word information to build the word vectors it can even produce vectors for words that did not appear in the training data, such as misspelled words or concatenations of words.

The model used in these experiments is the text feature extraction part of the proposed model, as shown in Figure 2, with an added softmax function for the predictions after the fully connected layer. We use standard text pre-processing, including lowercase and removal of special characters before the tokenization. We also use a Logistic Regression model as a baseline for the comparisons, as it is a well-known vastly used traditional machine learning algorithm. The input of the Logistic Regression is a traditional bag-of-words model created by the Count Vectorizer and for both of them we use the default parameters of scikit-learn.

Table II shows the detailed results of the experiments on both datasets. For all methods we provide the accuracy of the fake news detection, as well as the precision, recall and F1 scores for both fake and real news. In Table III we also show the vocabulary coverage of both datasets for all pre-trained word embeddings. PolitiFact is a much smaller dataset than GossipCop, so it also has a smaller vocabulary size, however, we can see that the GloVe embeddings provide a greater vocabulary coverage for both datasets, followed by FastText and lastly Word2Vec.



(a) Word2Vec (b) GloVe (c) FastText

(d) Word2Vec (e) GloVe (f) FastText

Fig. 3: Visualization of different word embeddings corresponding to the vocabulary of each dataset. (a, b, c) PolitiFact. (d, e, f) GossipCop.

On the PolitiFact dataset the CNN cannot perform much better than the Logistic Regression, as the dataset is too small and the neural network cannot learn so effective latent text features. However, it is clear that by using the CNN with the FastText embeddings we can get higher quality results for both fake and real news, based on the accuracy and F1 scores.

On the GossipCop dataset, we can see that the CNN performs much better than the Logistic Regression as it can learn from more data. Also, it is interesting that even though the GloVe embeddings cover 65% of the vocabulary and FastText only 58% their performance is quite similar, with an almost

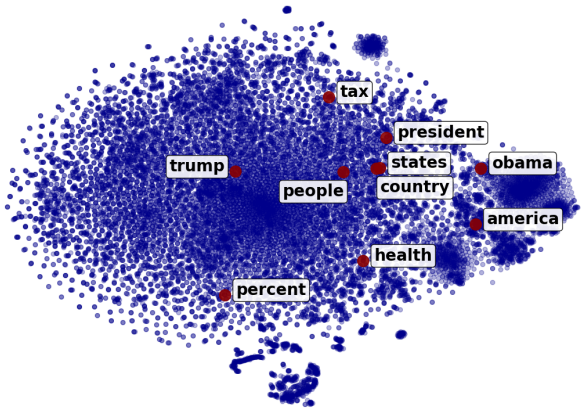
TABLE III
VOCABULARY COVERAGE OF DIFFERENT WORD EMBEDDINGS

Dataset	Vocabulary	Word2Vec	FastText	GloVe
PolitiFact	23130	75%	84%	86%
GossipCop	84077	46%	58%	65%

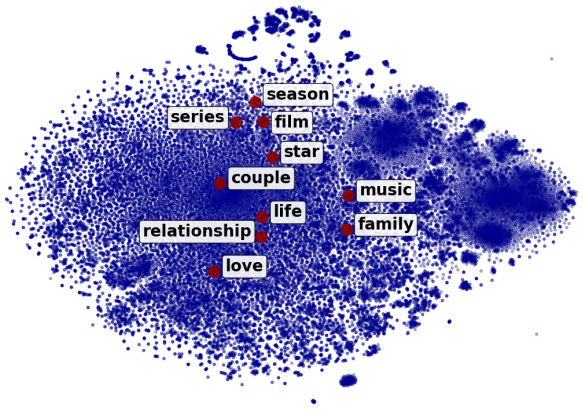
equal accuracy and the FastText performing a bit better on the prediction of fake news.

By using the FastText embeddings we got slightly better results, while the Word2Vec performed worse on both datasets compared to FastText and GloVe. Therefore, we decided to use FastText pre-trained vectors for our proposed model. In order to understand the performance difference of the embeddings we visualised them in Figure 3 using the t-SNE algorithm for dimensionality reduction initiated with PCA for more accurate results. It is visible that FastText and GloVe embeddings found more and denser clusters of similar words compared to Word2Vec. Thus, using GloVe and FastText embeddings helped to extract more meaningful and accurate features.

In Figure 4 we show some of the most frequent words of both datasets mapped on FastText embeddings. For example, the words "states" and "country" are very similar and as a result they are extremely close to each other. Similarly, the words "film", "season" and "series" have a very small distance between them as they are often mentioned in the same context. By examining the frequent words, it is also clear that the news in PolitiFact are mostly related to politics, while GossipCop contains news stories about the personal lives of celebrities and about entertainment like TV series and movies.



(a) PolitiFact



(b) GossipCop

Fig. 4: Visualization on FastText embeddings of some of the most frequent words of both datasets.

C. Visual model performance

In this subsection, we introduce the state-of-the-art deep learning models used for the extraction of features from images and we compare their performance for the detection of fake news using only the visual content. All models are pre-trained on the ImageNet dataset and are available through Keras applications.

1) **VGG-19** [9]: VGG-19 is a Convolutional Neural Network consisting of 19 layers and uses convolution filters of size 3x3. We extract the image features from the last 4096-dimensional fully connected layer.
2) **ResNet-152 V2** [10]: ResNet-152 V2 is a Residual Neural Network that consists of many stacked Residual Units with a total of 152 layers. It uses identity mappings as the skip connections to jump over layers and identity after-addition activation for making information propagation smooth. We extract the image features from the last 2048-dimensional average-pooling layer.
3) **MobileNet V2** [11]: The MobileNet V2 architecture is based on an inverted residual structure where the input and output of the residual block are thin bottleneck layers opposite to traditional residual models which use expanded representations. We extract the image features from the last 1280-dimensional average-pooling layer.
4) **DenseNet-201** [12]: The DenseNet-201 (Dense Convolutional Network) connects each layer to every other layer in a feed-forward fashion unlike traditional Convolutional Networks that have one connection between each layer and its subsequent layer. We extract the image features from the last 1920-dimensional average-pooling layer.
5) **Inception V3** [13]: Inception V3 is a Convolutional Neural Network with factorized convolutions and aggressive dimension reductions that uses batch-normalized auxiliary classifiers and label-smoothing. We extract the image features from the last 2048-dimensional average-pooling layer.
6) **NasNet Large** [14]: NasNet searches for an architectural building block on a small dataset and then transfers the block to a larger dataset by stacking together more copies of this block, each with their own parameters to

TABLE IV
PERFORMANCE COMPARISON OF MODELS USING ONLY VISUAL CONTENT

Dataset	Model	Accuracy	Fake News			Real News		
			Precision	Recall	$F1$	Precision	Recall	$F1$
PolitiFact	VGG-19	0.765	0.720	0.783	0.750	0.808	0.750	0.778
	ResNet-152 V2	0.765	0.729	0.761	0.745	0.796	0.768	0.782
	MobileNet V2	0.755	0.733	0.717	0.725	0.772	0.786	0.779
	DenseNet-201	0.775	0.745	0.761	0.753	0.800	0.786	0.793
	Inception V3	0.794	0.766	0.783	0.774	0.818	**0.804**	0.811
	NasNet Large	0.794	0.755	0.804	0.779	0.830	0.786	0.807
	Xception	**0.804**	**0.771**	**0.804**	**0.787**	**0.833**	0.803	**0.818**
GossipCop	VGG-19	0.594	0.582	0.632	0.606	0.608	0.557	0.581
	ResNet-152 V2	0.601	0.592	0.621	0.606	0.612	0.583	0.597
	MobileNet V2	0.608	0.598	0.630	0.614	0.619	0.587	0.603
	DenseNet-201	0.616	0.607	0.630	0.618	0.625	0.602	0.614
	Inception V3	0.623	0.616	0.627	0.622	0.630	**0.619**	0.624
	NasNet Large	0.629	0.618	0.654	0.635	0.642	0.606	0.623
	Xception	**0.633**	**0.622**	**0.654**	**0.638**	**0.645**	0.613	**0.629**

design a convolutional architecture. We extract the image features from the last 4032-dimensional average-pooling layer.

7) **Xception** [15]: Xception uses a Convolutional Neural Network architecture inspired by Inception, where Inception modules have been replaced with depthwise separable convolutions (a depthwise convolution followed by a pointwise convolution). We extract the image features from the last 2048-dimensional average-pooling layer.

Table IV shows the results of the experiments on both datasets, including the accuracy, precision, recall and F1 scores for each model used in image feature extraction. For our experiments we use the architecture described as the visual feature extraction part of our proposed model in Figure 2, i.e, we pass the output of each deep learning model through a 32-dimensional fully connected layer and then we add a softmax function to classify news into fake or real. All images are first resized and pre-processed according to each model's specifications.

Using the Xception model for extracting features from images produced the best results on both datasets, therefore we included it in our proposed model architecture. NasNet Large and Inception V3 can also extract high quality features from images and we can see that the models produce similar results compared to each other on both datasets, with the exception that on the GossipCop dataset, VGG-19 has the lowest accuracy, while on PolitiFact, MobileNet V2 scored the lowest. The choice of the model for the image feature extraction is important in order to acquire higher quality predictions. In fact, on GossipcPop there is a difference of 0.04 in the accuracies of Xception and VGG-19, and on PolitiFact the difference between Xception and MobileNet V2 is 0.05.

(a) PolitiFact fake news images

(b) PolitiFact real news images

(c) GossipCop fake news images

(d) GossipCop real news images

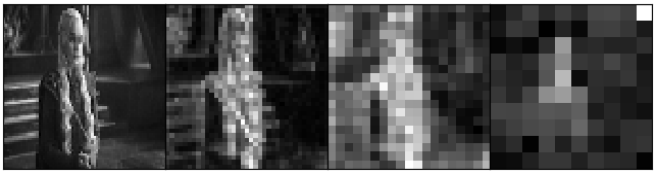
Fig. 5: Example images in fake and real news of both datasets.

The performance of fake news detection using only the visual content of news differs significantly between the two datasets. On PolitiFact we get an accuracy of 0.804 using Xception, while the same model results to an accuracy of 0.633 on GossipCop. The difference of size between the two

TABLE V
PERFORMANCE COMPARISON OF THE SINGLE-MODAL MODELS AND THE PROPOSED MULTI-MODAL MODEL

Dataset	Model	Accuracy	Fake News			Real News		
			Precision	Recall	$F1$	Precision	Recall	$F1$
PolitiFact	Text	0.814	0.787	0.804	0.796	0.836	0.821	0.829
	Vis	0.804	0.771	0.804	0.787	0.833	0.803	0.818
	Text+Vis	**0.853**	**0.860**	**0.804**	**0.831**	**0.847**	**0.893**	**0.870**
GossipCop	Text	0.769	0.782	0.738	0.759	0.758	0.799	0.778
	Vis	0.633	0.622	0.654	0.638	0.645	0.613	0.629
	Text+Vis	**0.781**	**0.801**	**0.740**	**0.769**	**0.764**	**0.821**	**0.791**

(a) (b)



(c)

Fig. 6: Visualization of an image's feature maps extracted from different convolution layers of the Xception model. (a) Input image. (b) First six feature maps of the first convolution layer. (c) Selected feature maps of four intermediate convolution layers.

datasets surely affects the results to some extent, but by taking a closer look to the distribution patterns of images we can better understand these results. Figure 5 shows some indicative example images in fake and real news of both datasets. The PolitiFact dataset, mostly contains images of politicians. However, the images of real news are mostly related to specific real events, while the images of fake news can be more abstract and diverse. As a result, it is easier to distinguish patterns unique to fake or real news images and produce higher quality predictions. On the other hand, almost all images on GossipCop are photos of people as the news are related to celebrities. Therefore, it is harder to extract features that can accurately distinguish an image that belongs to a fake news story from an image of a real news story. The visual model, however, can still learn to classify the images. For example, a difference between the fake and real news images in GossipCop is that the fake news contain more photos of celebrities shot outside of official context, like from paparazzi.

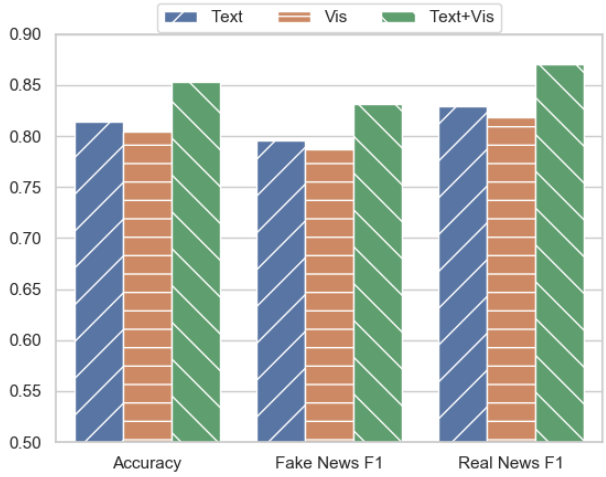
The features extracted from deep learning models are not as easy to understand as hand-crafted features extracted from images [29]. Figure 6 shows the internal representation of an input image of the Xception model in order to take a closer look at what kind of features this model can learn. The first convolution layer of Xception produces 32 feature maps and we show the first six of them, where we can see how the image is transformed with different filters applied to it. We also selected one feature map from each of four intermediate convolution layers to show how the image is represented in deeper parts of the network. It is visible that the feature maps close to the input detect small details, while feature maps close to the output of the model capture more general features. After applying the average-pooling operation this kind of general features are used to represent images of fake and real news. Then, a simple fully connected layer and softmax function are used to discriminate fake from real images based on these representations obtained by the deep learning model.

D. Proposed model performance

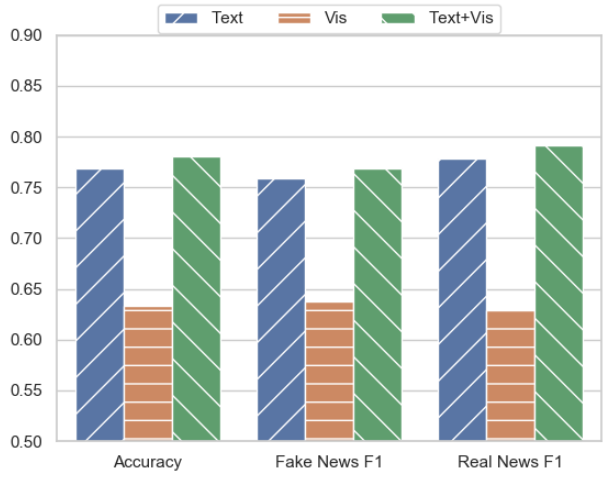
In this subsection, we present the results of the proposed model that combines features extracted from both text and images to detect fake news and we compare it to the models that use only text content or only visual content.

Table V shows the results of fake news detection using different modalities. The Text model refers to the CNN using FastText pre-trained word embeddings, as we found that they can achieve better results and the Visual model refers to using Xception for the image feature extraction, which was the best among the models we experimented with. Finally, Text+Vis is the proposed model, which combines the Text and Visual models in order to detect fake news using both text and image features.

On the PolitiFact dataset the Text and Visual models have similar performance in fake news detection. The proposed model in this case can effectively combine them and achieve significantly better results. On the other hand, the Visual model does not perform as well on The GossipCop dataset.

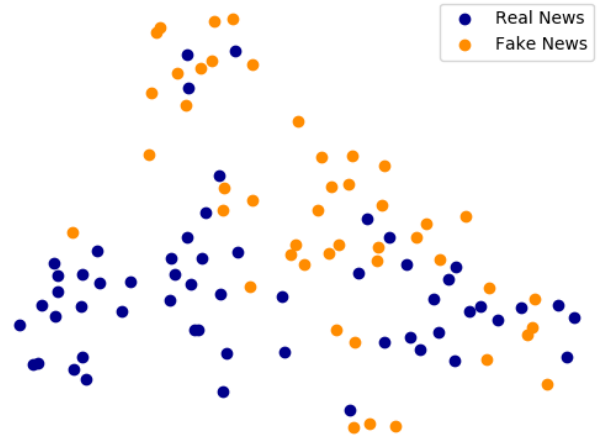


(a) PolitiFact

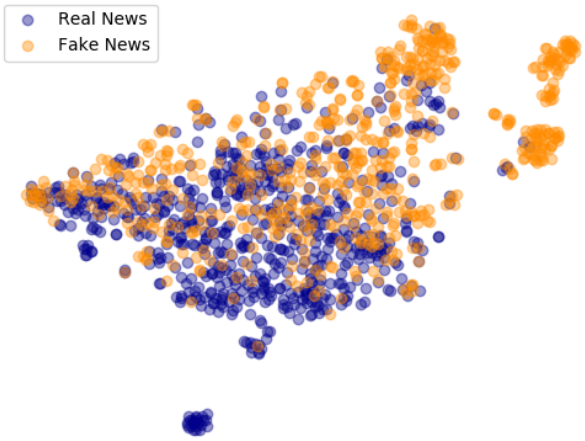


(b) GossipCop

Fig. 7: Effect of combining text and visual content.



(a) PolitiFact



(b) GossipCop

Fig. 8: Visualization of learned latent text feature representations on the testing sets of both datsets.

This poor performance affects the results of the proposed model, however, it can still produce slightly better results by combining the text and image features. Therefore, we can see that even in the case where the classification of news using only images is not as successful as the classification using text content, the proposed model can still improve the predictions. Figure 7 shows the results in terms of accuracy and F1 scores and we can observe the difference in performance of the proposed model compared to the single-modal models on both datasets.

The fake news detection is more successful on the PolitiFact dataset. The high performance of the visual model plays a major role in that regard, but the fake news can also be better distinguished from real news based on text content. Figure 8 shows the t-SNE based represantation of the latent text features learned by the proposed model on the tesing sets of both datasets. Specifically, we visualize the output of the global max-pooling operation. We can observe that on the GossipCop dataset there is a lot of overlapping between fake and real news features, while on the PolitiFact they are better discriminated and thus, the model performs better.

V. Conclusions

In this work, we study the problem of detecting fake news using both the text and visual content. We propose a deep learning model that can extract and combine text and image features in order to improve the performance of real-time fake news detection. For that purpose, we compared different word-embeddings and state-of-the art deep learning models for image feature extraction, to learn effective latent representations of the news content. Extensive experiments on two real-world datasets demonstrate the effectiveness of our model. The results show that by using visual content, the detection of fake news can be improved, but the degree of this improvement is affected by the visual distributions of each dataset.

REFERENCES

[1] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. Defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 395–405, New York, NY, USA, 2019. Association for Computing Machinery.

[2] Xinyi Zhou and Reza Zafarani. Fake news: A survey of research, detection methods, and opportunities, 2018.

[3] Xinyi Zhou, Atishay Jain, Vir V. Phoha, and Reza Zafarani. Fake news early detection: A theory-driven model, 2019.

[4] Zhiwei Jin, Juan Cao, Yongdong Zhang, Jianshe Zhou, and Qi Tian. Novel visual and statistical image features for microblogs news verification. *IEEE Transactions on Multimedia*, 19(3):598–608, March 2017.

[5] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 849–857, New York, NY, USA, 2018. Association for Computing Machinery.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA, 2013. Curran Associates Inc.

[7] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[8] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision – ECCV 2016*, pages 630–645. Springer International Publishing, 2016.

[11] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 2018.

[12] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, July 2017.

[13] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016.

[14] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 2018.

[15] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, July 2017.

[16] Lawrence Boehm. The validity effect: A search for mediating variables. *Personality and Social Psychology Bulletin*, 20:285–293, 06 1994.

[17] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[18] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 3818–3824. AAAI Press, 2016.

[19] Tong Chen, Lin Jung Wu, Xue Li, Jun Zhang, Hongzhi Yin, and Yang Wang. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. *ArXiv*, abs/1704.05973, 2017.

[20] Ke Wu, Song Yang, and Kenny Q. Zhu. False rumors detection on sina weibo by propagation structures. *2015 IEEE 31st International Conference on Data Engineering*, pages 651–662, 2015.

[21] Kai Shu, Xinyi Zhou, Suhang Wang, Reza Zafarani, and Huan Liu. The role of user profiles for fake news detection. *ArXiv*, abs/1904.13355, 2019.

[22] Kai Shu, Suhang Wang, and Huan Liu. Beyond news contents. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining - WSDM*. ACM Press, 2019.

[23] François Chollet et al. Keras. https://keras.io, 2015.

[24] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[25] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[26] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *arXiv preprint arXiv:1809.01286*, 2018.

[27] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36, 2017.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[29] Dongping Tian. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, 8:385–395, 01 2013.