INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

Mini Project 2020-21

A Report on

Multimodal Fake News Detection.

Under the guidance of

Dr. Sunil Saumya
Assistant Professor, Dept. of CSE,
IIIT Dharwad.

Team (Group - 13):

Shaik Fharook[1], 3rd Year, Dept. of CSE,
IIIT Dharwad, Regd. No: 18BCS091.

Gurram Rithika[2], 3rd Year, Dept. of CSE,
IIIT Dharwad, Regd. No: 18BCS031.

Pravalika[3], 3rd Year, Dept. of CSE,
IIIT Dharwad, Regd. No: 18BCS070.

Godina Pranav[4], 3rd Year, Dept. of CSE,
IIIT Dharwad, Regd. No: 18BCS028.

I. Abstract

Nowadays, information can easily be accessible from anywhere. It is the age of information, where an individual can access the happenings of various events around the world in the comfort of his/her own home. It has resulted in the inaccuracy and irrelevancy in updating information by people which is commonly known as fake news. Since a large proportion of the population uses social media for updating themselves with news, delivering accurate and altruistic information to them is of utmost importance. Due to the increasing number of users in social media, news can be quickly published by anyone, and chances are high that the credibility stands of information can be easily compromised. As fake news is written to mislead readers, it makes a difficult task to detect just based on the content of the news. Fake news detection has recently garnered much attention from researchers and developers alike. The news content is diverse in terms of styles, the subject in which it is written made current existing deep learning models more difficult in judging/detecting the fake news. This leads an essential to bring an efficient system for its detection considering multiple modalities both image and text. In this paper, we've implemented a well-known multimodal based algorithm called ***TI-CNN: Convolutional Neural Networks for Fake News Detection*** [1] (proposed by Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Xiaoming Zhang, Zhoujun Li, Phillip S. Yu) along with different other existing simple CNN based algorithms & state-of art algorithms which includes ResNet-50 & LSTM.

I. Introduction

With the rapid advancement in the technology, a large number of experiments continue to be conducted in order to solve problems which were never considered in the context of computer science. One such problem is that of fake news detection. Since the access to news media has become very convenient, as soon as some noteworthy event occurs anywhere around the globe, different news sources tend to make their news eye-catching in order to deliver news to as many people as possible on the worldwide web. This in turn results in the quick dissemination of news to millions and billions of people through a large number of news sources such as news channels, articles, websites, and social networking sites.

Apart from several reputable news organizations and agencies which have been operating on an international level for decades, and deliver news to the general public, there are a large number of smaller news sources which deliver news that are not trustworthy. In addition to this, in popular social networking and social media platforms anyone from anywhere around the globe can publish and disseminate any kind of statement, or set of statements, to spread fake news through the use of different networking sites in order to achieve different goals, which may be fraudulent or illegal. A major caveat is that some of the sources that are considered to be authentic, and are popular sources for informational services, such as Wikipedia, are also prone to false information or fake news. In addition to this, because some official news aggregators may deliberately spread false or fake news in order to gain popularity, achieve some political objective, or earn money, the problem is further intensified. Another factor contributing to the spread of fake news may be organized astroturfing campaigns which attempt to mock or spoil a specific product or company, a society or a group of people, e.g. for political, social, or financial reasons.

Fake news is considered to be one of the greatest threats to commerce, journalism and democracy all over the world, with huge collateral damages. A US $130 billion loss in the stock market was the direct result of a fake new report that US president Barack Obama got injured in an explosion. Other cases of fake news campaigns that demonstrate the enormous impact that fake news can have include the sudden shortage of salt in Chinese supermarkets after a fake report that iodized salt would help counteract the effects of radiation after the Fukushima nuclear leak in Japan, and an escalation of tensions between India and Pakistan that began with fake reporting of the Balakot strike and resulted in the deaths of military personnel and the loss of expensive military equipment.

The term ‘fake news’ is often described in related literature as ‘misinformation’, ‘disinformation’, ‘hoax’, and ‘rumour’, which are actually different variations of false information. There are a variety of research projects, tools and applications for fact checking, and fake news detection, which mostly examine the problem as a veracity classification.

The dissemination of fake news through different mediums, especially online platforms, has not been stopped completely or scaled down to a degree in order to reduce the adverse effects fake news can lead to. The reason is that there is no system that exists that can control fake news with little or no human involvement. Experiments indicate that machine and learning algorithms may have the ability to detect fake news, given that they have an initial set of cases to be trained on.

Deep learning techniques have great prospects in fake news detection tasks. There are very few studies suggesting the importance of neural networks in this area. The model implemented is the hybrid neural network model which is a combination of convolutional neural networks for both image and text. As this model is required to classify between fake news and legitimate news.

II. Related Works

Majority of previous research done at the news detection level was heavily dependent on text and user metadata features. Potthast et al. [2] showed how writing style, network connection and user reaction can lead to the detection of fake news. Moreover, Shu et al. [3] described how the writing style of an author impacts the views and opinions of people reading such articles. This plays a vital role in shaping the opinions of the masses. To improve fact analysis in news content, Pan et al. [4] used knowledge graphs. Entity relation information extracted out of these graphs can be used to induce common sense reasoning into text content. Recently, Lin et al. [5] used TransR model to generate knowledge graph embeddings (KGE) for the entity relation triplets extracted from news articles. The advantage of using TransR to get KGE is that, it builds entity and relation embeddings in distinct spaces. KGE learning is then done by projecting entities from entity space to corresponding relation space and then building translations between projected entities. Forging fake images is a popular way to tamper news. To detect such incidents image splicing technique [6] was used that takes input as the EXIF metadata information and determines whether the image is self-consistent or not. Recently, Marra et al. [7] used GANs to detect fake images. Though all the above mentioned uni-modal techniques were able to provide promising results, short and informal nature of social media data always becomes a challenge in information extraction. To overcome this limitation, the researchers started experimentation with features extracted from multiple modalities (i.e. text and image) and fused them together for richer data representation. Works [8], [9], [10] are the most notable studies in multimodal fake news detection.

Though these multimodal systems perform well in detecting fake news, the classifiers have always been trained in tandem with another classifier. This increases training and model size overhead, increases training complexity and at times can also hinder the generalizability of the systems due to lack of data for the secondary task.

To solve such issues, we design a multimodal framework for fake news detection. It takes into consideration features from two different modalities and classifies the sample into real or fake without considering any other sub-task. Next, we highlight the details of the multimodal.

III. Methodology.

i. Dataset Description.

In this project we consider using a well-known dataset called ***Fakeddit: A New Multimodal Benchmark Dataset for Fine-grained Fake News Detection*** [11] *(proposed by Kai Nakamura, Sharon Levy, William Yang Wang)*. This is a novel multimodal dataset consisting of over 1 million samples from multiple categories of fake news. The samples in this dataset is extracted from one of the social media platforms called *Reddit.* Each sample in this dataset is processed through several stages and are labelled as 2-way, 3-way and 6-way classification categories. Each sample contains an image and text information along with all the relevant information i.e., the author name, metadata, comment data etc.

Data	**No. of Samples(approx)**	**2-Way**	**3-Way**	**6-Way**
Training	60 lakhs	True	True	True
Test	60 thousand	True	True	True
Validate	60 thousand	True	True	True

ii. Dataset Fact-Checking.

As the main motive of every machine learning mode/ deep learning model is to predict good accuracy on the unseen data, it's very much essential to fact check the training data as the model performance will rely on it. The authors of the dataset [11] have presented the transparency in selecting the samples. *Fakeddit* can also be applied to the realm of implicit fact-checking. Other existing datasets utilized for fact-checking include FEVER (Thorne et al., 2018) and Fauxtography (Zlatkova et al., 2019). The former consists of altered claims utilized for textual verification. The latter utilizes both text and image data in order to fact-check claims about images. Using both text and image data, researchers can use Fakeddit for verifying truth and proof: utilizing image data as evidence for text truthfulness or using the text data as evidence for image truthfulness. Compared to other existing datasets, Fakeddit provides a larger breadth of novel features that can be applied in a number of applications: fake news text, image, text + image classification as well as implicit fact-checking. Other data provided, such as

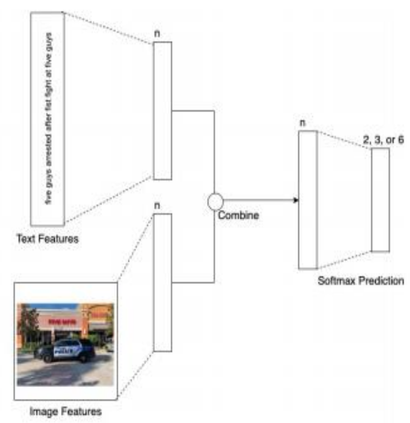
Dataset	Size (# of samples)	# of Classes	Modality	Source	Data Category
LIAR	12,836	6	text	Politifact	political
FEVER	185,445	3	text	Wikipedia	variety
BUZZFEEDNEWS	2,282	4	text	Facebook	political
BUZZFACE	2,263	4	text	Facebook	political
some-like-it-hoax	15,500	2	text	Facebook	scientific/conspiracy
PHEME	330	2	text	Twitter	variety
CREDBANK	60,000,000	5	text	Twitter	variety
Breaking!	700	2,3	text	BS Detector	political
NELA-GT-2018	713,000	8 IA	text	194 news outlets	variety
FAKENEWSNET	602,659	2	text	Twitter	political/celebrity
FakeNewsCorpus	9,400,000	10	text	Opensources.co	variety
FA-KES	804	2	text	15 news outlets	Syrian war
Image Manipulation	48	2	image	self-taken	variety
Fauxtography	1,233	2	text, image	Snopes, Reuters	variety
image-verification-corpus	17,806	2	text, image	Twitter	variety
The PS-Battles Dataset	102,028	2	image	Reddit	manipulated content
Fakeddit (ours)	**1,063,106**	**2,3,6**	**text, image**	**Reddit**	**variety**

Table 1: Comparison of various fake news detection datasets. IA: Individual assessments.

comments data, enable more applications. The Table -1 shows the comparison of Fakeddit with different available datasets both multimodal and unimodal and reflects the diversity and uniqueness of dataset.

iii. Basic Methodology.

Based on one of the survey results and previous literature, it is evident that a multimodal system is necessary for fake news detection. However, we wanted our system to be able to detect fake news independently without any other subtask, as seen in the current state-of-the-art systems. The fake news classifier of the current state-of-the-art system does not perform well by itself. However, performance significantly improves in the presence of a secondary task like sample reconstruction. To this end, we propose a multimodal framework for fake news detection. Our multimodal is divided into three sub-modules. The first sub-module is a textual feature extractor that extracts the contextual text features using a language model. The second sub-module is a visual feature extractor that extracts the visual features from a post. Finally, the last sub-module is a multimodal fusion module that combines the representations obtained from different modalities together to form a news feature vector. The complete outline of our multimodal is shown in the below figure.



iv. Custom Dataset Extraction *(Extracting 10%, 20% of dataset.)*

Since the dataset is very huge (106 GB training image data), it's highly impossible to run our models on entire dataset with limited resources. For our convenience, we developed an automated python script which'll extract the desired percent of dataset we want. The script also maintains the skewness of the dataset extracting equal amounts of true and false samples. Keeping mind, the limited resources, we've extracted 10%of the dataset which consists of 56400 training samples, 5940 testing samples and 5940 validating samples.

v. Data Pre-processing

It's a common practise to pre-process the data before passing it to the model for training / validation. In this stage, different types of noise is/are removed and normalized for the best performance of the model. As our multimodal framework depends on two factors, image and text, all the samples are pre-processed separately.

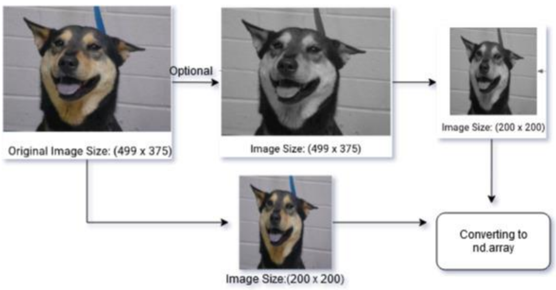
It's usual that images downloaded from internet have high chances of corruption and it's very important to detect and repair such images to maintain the quality standards of data. As a part of it, we developed a python script which detects the corrupted images. Surprisingly, we discovered a little amount of sample data is corrupted (<1000) which is insignificant when compared to the total 100% dataset. (≈ 60 lakh+). As a part of data cleaning, we removed the corrupted data samples from the dataset.

A. *Image Pre-Processing.*

Basically, an image is a 2-dimensional matrix stretched in 3 channels referred as *red, green, blue (rgb)*. The common standard of image pre-processing technique is divided into 3 phases.

- Reading an image.
- Converting image to grayscale (optional).
- Resizing the image to a pre-defined size.

We used *OpenCV,* an opensource python-based computer vision library to read, convert and resize images. The process can be picturised in the following figure.



B. *Text Pre-Processing*

There are numerous tools available for automating much of this pre-processing and text data preparation, however. These tools existed prior to the publication of those articles for certain, but there has been an explosion in their proliferation since. Since much NLP work is now accomplished using neural networks, it makes sense that neural network implementation libraries such as TensorFlow and also, yet simultaneously, Keras would include methods for achieving these preparation tasks.

Here we will look at tokenizing and further preparing text data for feeding into a neural network using TensorFlow and Keras pre-processing tools. There are 3 stages of text pre-processing:

a) *Tokenization*

First, we create the Tokenizer object, providing the maximum number of words to keep in our vocabulary after tokenization. The below picture shows how a text is tokenized.

```
(['this girl on a wheelchair is fighting for the rights of children with disabilities globally',
  'zermatt', 'frostbite the mountaineers', ...,
  'the bringer of the apocalypse', 'two cats licking glass door',
  'forbes how commissioner carr can modernize the fcc'], dtype=object)
```

b) *Encoding Sentences to Sequences.*

Now that we have tokenized our data and have a word to numeric representation mapping of our vocabulary, let's use it to encode our sequences. Here, we are converting our text sentences from words to integers.

```
Training sequences:
 [[4, 237, 238, 239, 240, 5, 2, 241, 6, 4, 108], [242, 35, 243, 36, 3, 244], [245, 35, 246, 37, 247, 109, 248, 249, 110, 11, 109], [250, 251, 2, 252],
```

c) *Padding Sequences*

We need our encoded sequences to be of the same length. We just found out the length of the longest sequence, and will use that to pad all other sequences with extra '0's at the end.

```
Padded training sequences:
 [[   4  237  238 ...    0    0    0]
 [ 242   35  243 ...    0    0    0]
 [ 245   35  246 ...    0    0    0]
 ...
 [   4 1111 1112 ...    0    0    0]
 [1113 1114 1115 ...    0    0    0]
 [   4 1121 1122 ...    0    0    0]]
```

vi. *Model Description.*

In this section, we'll look into 7 different models of which 6 are unimodal and 1 multimodal deep learning frameworks. The performance of models also justifies the importance of multimodal frameworks in this scenario such as fake news detection.

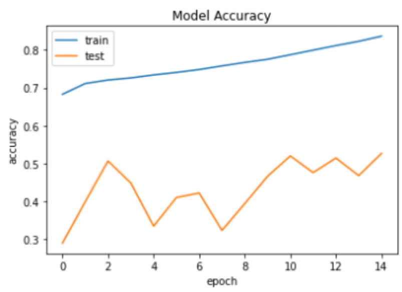
A. Simple CNN: Image

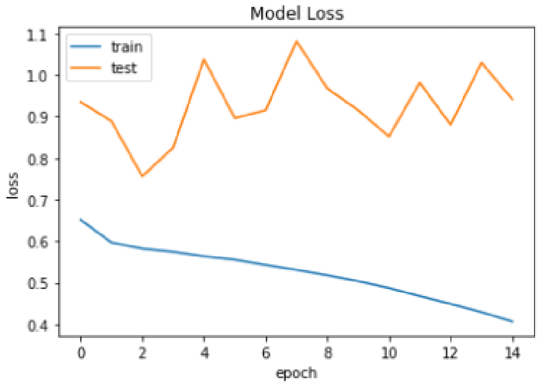
The following model is a simple CNN for image classification. L2 Regularization is applied at each Conv2D layer. The image is converted into grayscale to reduce the dimensionality which'll eventually reduce the size of image array by times. The below table shows the weights used at each layer.

Simple CNN: Image	
Layer Name	**Weights**
Input Layer	(120 x 120 x 1)
Conv2D	filters=64, kernel_size= (3,3)
Activation	Relu
MaxPooling2D	pool_size= (2,2)
Conv2D	filters=64, kernel_size= (3,3)
Activation	Relu
MaxPooling2D	pool_size= (2,2)
Flatten	
Dense	128
Dense	1
Activation	Sigmoid

After training on 56400 samples for 15 epochs, the simple CNN model for image achieved an accuracy of 82% of training accuracy and 57% of test / validation accuracy.

The below pictures depict the graphical representation of the model's accuracy and model loss respectively.





B. Transfer Learning (ResNet-50)

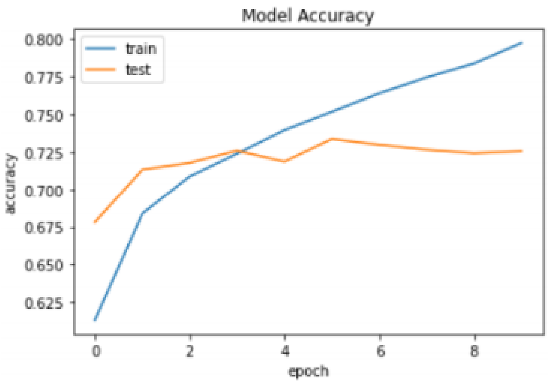
When comes to better feature extraction of images, state-of-art algorithms will give very good results as they are highly trained on more than a million of different set of images. One such algorithms is ResNet-50, an effective 50-layered pre-trained model.

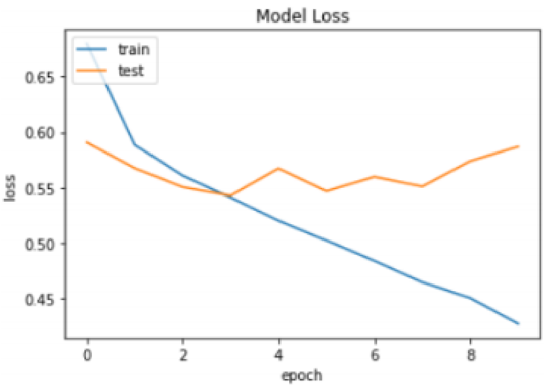
ResNet-50 is based on Residual Neural Network which follows a technique called **'skip connection'** unlike the traditional conv networks to avoid the problem of vanishing gradient. It contains 48 Convolutional layers along with 1 MaxPool and 1 Average Pool Layer. In addition to that the architecture has $3.8 * 10^9$ floating points operations which are really great at identifying low, mid and high-level features from the images.

The table below shows the weights used at each layer.

Transfer Learning (ResNet-50)	
Layer	**Weights**
Input Layer	120 x 120 x 3
ResNet-50	all_layers = Trainable
Dense	512
Activation	Relu
Dropout	0.3
Dense	512
Activation	Relu
Dropout	0.3
Dense	1
Activation	Sigmoid

After training on 56400 samples for 10 epochs, the Transfer learning model for image classification achieved an accuracy of 80% of training accuracy and 72.5 % of test / validation accuracy. The below pictures depict the graphical representation of the model accuracy and model loss respectively.





The results justify that Resnet-50 model shows its best performance in extraction features of images and gives good results in classification of images.

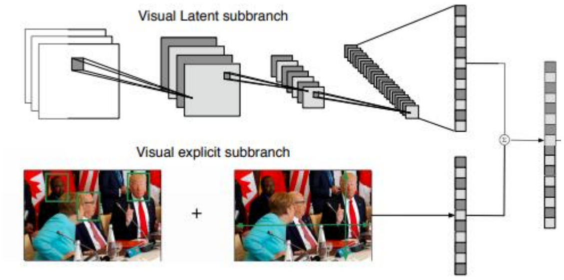
C. CNN + Dense: Image

This model is a part of algorithm taken from the research paper called ***TI-CNN: Convolutional Neural Networks for Fake News Detection*** [11], which proposes an additional feature extraction branch called Visual Explicit. According to the research paper, the image features are extracted in two way one the traditional visual latent branch and the other visual explicit branch. The Visual latent branch extracts the features of an image by different sets of Convolution, MaxPooling layers along with Dropout, Batch Normalization and Dense layers. Whereas, the visual explicit branch is built up of pure Dense layers along with Batch Normalization which looks for the explicit features the image.

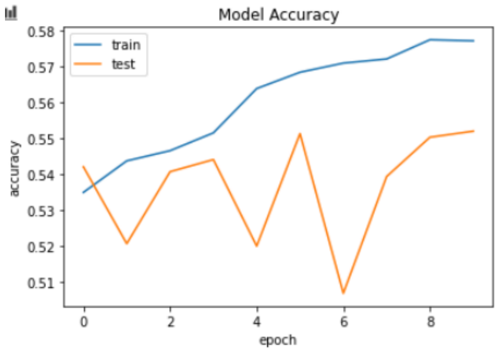
The outputs of both the visual latent and explicit are merged and passed to the dense neural network for classification.

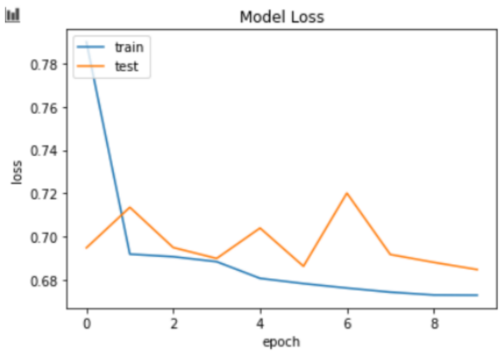
<table>
<tr><th colspan="4">CNN + Dense: Image</th></tr>
<tr><th colspan="2">Image Latent</th><th colspan="2">Image Explicit</th></tr>
<tr><th>Layer Name</th><th>Weights</th><th>Layer Name</th><th>Weights</th></tr>
<tr><td>Input Layer</td><td>120 x 120 x 3</td><td rowspan="3">Input Layer</td><td rowspan="3">120 x 120 x 3</td></tr>
<tr><td>Conv2D</td><td>filters=32, kernel_size=(2,2)</td></tr>
<tr><td>Activation</td><td>Relu</td></tr>
<tr><td>Dropout</td><td>0.8</td><td rowspan="4">Dense</td><td rowspan="4">128</td></tr>
<tr><td>MaxPooling2D</td><td>pool_size= (2,2)</td></tr>
<tr><td>Conv2D</td><td>filters=32, kernel_size=(2,2)</td></tr>
<tr><td>Activation</td><td>Relu</td></tr>
<tr><td>Dropout</td><td>0.8</td><td colspan="2" rowspan="6">Batch Normalization</td></tr>
<tr><td>MaxPooling2D</td><td>pool_size= (2,2)</td></tr>
<tr><td>Conv2D</td><td>filters=32, kernel_size=(2,2)</td></tr>
<tr><td>Activation</td><td>Relu</td></tr>
<tr><td>Dropout</td><td>0.5</td></tr>
<tr><td>MaxPooling2D</td><td>pool_size= (2,2)</td></tr>
<tr><td colspan="2">Flatten</td><td rowspan="4">Activation</td><td rowspan="4">Relu</td></tr>
<tr><td>Dense</td><td>128</td></tr>
<tr><td colspan="2">Batch Normalization</td></tr>
<tr><td>Activation</td><td>Relu</td></tr>
<tr><td colspan="2">Merge</td><td colspan="2">Merge</td></tr>
<tr><td colspan="4">Merge</td></tr>
<tr><td colspan="2">Activation</td><td colspan="2">Relu</td></tr>
<tr><td colspan="2">Dense</td><td colspan="2">128</td></tr>
<tr><td colspan="4">Batch Normalization</td></tr>
<tr><td colspan="2">Dense</td><td colspan="2">1</td></tr>
<tr><td colspan="2">Activation</td><td colspan="2">Sigmoid</td></tr>
</table>

An illustration of CNN + Dense for image is shown in the below picture.



After training on 10% of dataset i.e., 56400 training samples for 10 epochs this model achieves an accuracy of 58% and 55% of test /validate accuracy. A detailed picture model's accuracy and loss can be visualized in the below pictures.



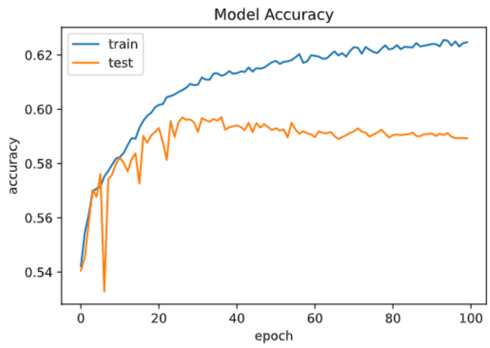


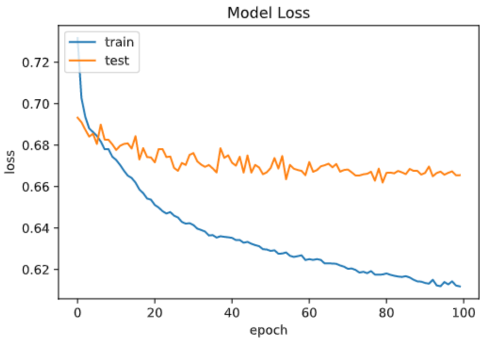
D. CNN based: Text

The following model is CNN based for text classification. The model is built up on 3 Conv1D layers along with MaxPooling1D, Dropout and Activation layers. The weights at each layer used at the experiment can be visualized in the following table.

CNN: Text	
Layer Name	**Dimension(s)**
Input Layer	(100 x 100)
Embedding	100 x 100
Conv1D	filters=16, kernel_size=4
Batch Normalization	pool_size=3
Activation	Relu
Dropout	0.5
Maxpooling1D	pool_size=2
Conv1D	filters=32, kernel_size=4
Batch Normalization	BN
Activation	Relu
Maxpooling1D	pool_size=2
Conv1D	filters=64, kernel_size=4
Batch Normalization	BN
Activation	Relu
Dropout	0.5
Maxpooling1D	pool_size=2
Flatten	
Dense	128
Dense	1

After training on 10% of the dataset i.e., 56400 training samples this model achieves an accuracy of 63% and 58% of text / validate accuracy. A detailed graph based of model's accuracy and loss can be visualised in the below picture.



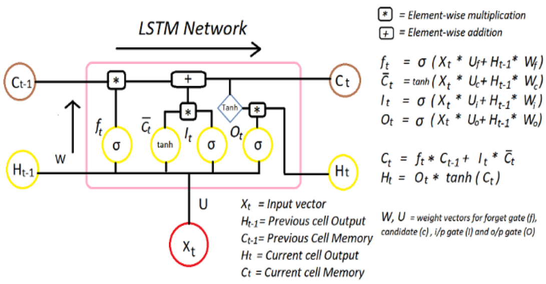


E. LSTM: Text

The following model is built up on Long short-term memory shortly LSTM, a very good algorithm especially for text classification jobs. LSTM is a type of artificial neural network architecture used to process multiple input data points in images, speech, audio as well as text. It consists of a cell and three gates, an input gate, forget gate and output gate.

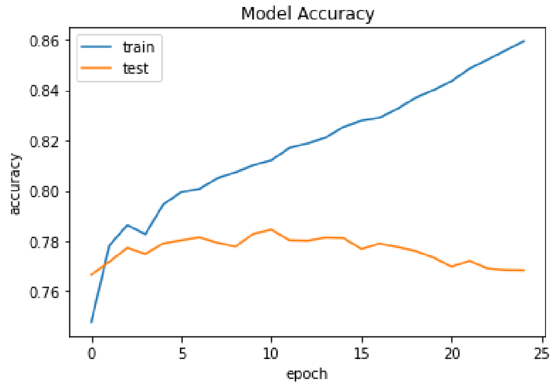
Unlike other architectures, LSTM has connections for feedback which are helpful regulating the information flow through the gates. The architecture is designed in such a way that it can remember the long-term dependencies of the data being presented to it. It could overcome the vanishing gradient problem that arises when using Recurrent Neural Networks. These models can be trained in both supervised and unsupervised manner.

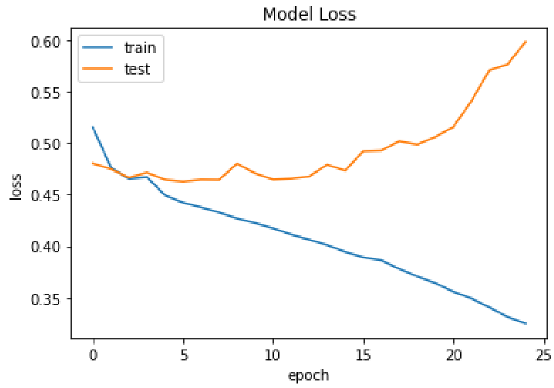
A simple workflow of LSTM can be visualised in the below picture and the weights used at each layer can be visualised in the following table.



LSTM: Text	
Layer Name	**Weights**
Input Layer	(100 x 100)
Embedding	100 x 50
LSTM	64
Dense	128
Activation	Relu
Dropout	0.5
Dense	1
Activation	Sigmoid

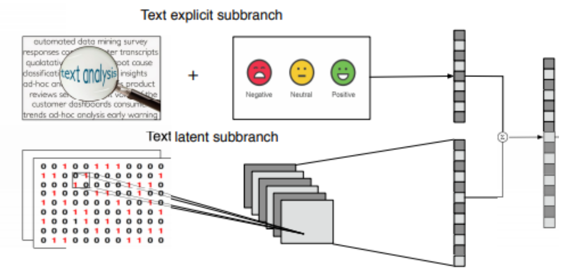
After training on 10% of dataset i.e., 56400 samples of training data for 25 epochs with early stopping, this model outrages an accuracy of 86% and 77% of validation accuracy. The following graphs represents the values of accuracy and loss at each epoch.





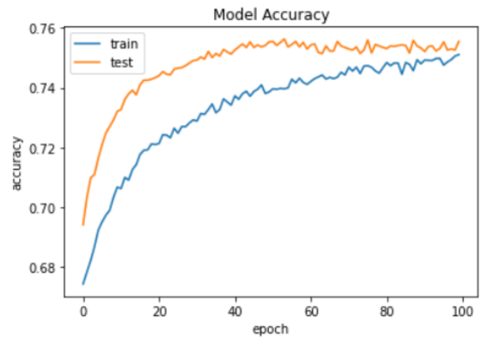
F. CNN + Dense: Text

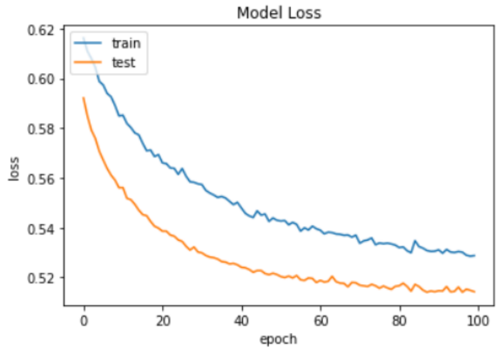
This model is a part of the algorithm taken from the research paper called ***TI-CNN: Convolutional Neural Networks for Fake News Detection*** [11], which proposes an additional feature extraction branch called Text Explicit. According to the research paper, the text features are extracted in two way, one the traditional text latent branch and the other text explicit branch. The Text latent branch extracts the features of the text data by different sets of Convolution, MaxPooling layers along with Dropout, Batch Normalization and Dense layers. Whereas, the text explicit branch is built up of pure Dense layers along with Batch Normalization which looks for the explicit features the text. The outputs of both the text latent and explicit are merged and passed to the dense neural network for classification.



CNN + Dense: Text			
Text Latent		**Text Explicit**	
Layer Name	**Weights**	**Layer Name**	**Weights**
Input Layer	100	**Input Layer**	100
Embedding layer	1000 x 100		
Dropout	0.5		
Conv1D	filters= 10, kernel_size=3	**Dense**	128
MaxPooling1D	pool_size=2		
Flatten			
Dense	128	**Batch Normalization**	
Batch Normalization			
Activation	Relu		
Dropout	0.8	**Activation**	Relu
Merge		**Merge**	
Merge			
Dense - 128			
Batch Normalization			
Dense - 1			
Activation - Sigmoid			

After training on 10% of data i.e., 54600 samples of training data the model for 100 epochs, the model shows an accuracy of 76% training accuracy and 75% of test / validate accuracy. The following graphs represents the values of accuracy and loss at each epoch.

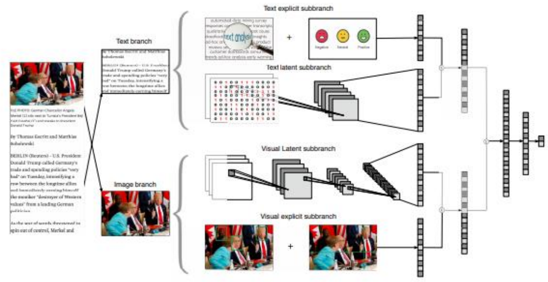




G. *TI-CNN: Multimodal (Text + mage)*

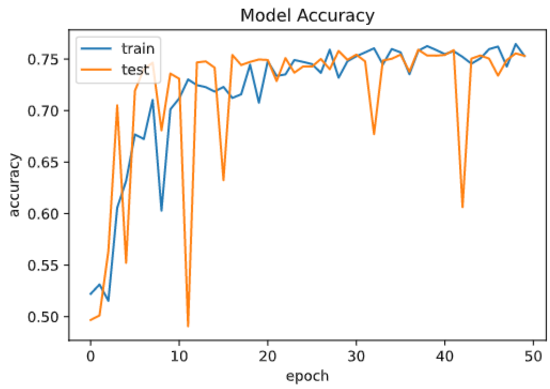
This model is taken from the research paper ***TI-CNN: Convolutional Neural Networks for Fake News Detection*** [11], one of the best algorithms which is built on Convolution layers for Fake News Detection multimodal. This algorithm can combine the text and image information with the corresponding explicit and latent features of both text and image which has the strong expandability, which can absorb most of the features of news. The Convolution layers in the model makes the model to see the entire input at a time and can show good results apart from the traditional unimodal models discussed before. The additional explicit layer for both image and text makes this model to extract more features apart from the information from convolution layers (latent branch).

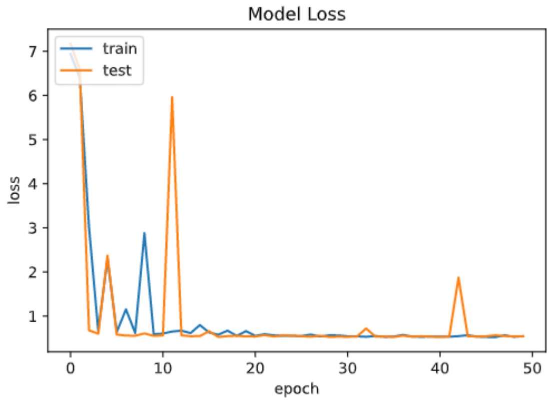
The following picture shows the work-



ing of the TI-CNN Model and the weights used at each layer can be visualised in the table TI-CNN: Image + Text.

After training on 10% of data i.e., for 56400 samples of data for 50 epochs this model achieves an accuracy of 75% of model accuracy and 75% accuracy of test / validate accuracy. The following graphs visualises the model's accuracy and loss at each epoch.





<table>
<tr><th colspan="8">TI-CNN : Image + Text</th></tr>
<tr><th colspan="4">Image Branch</th><th colspan="4">Text Branch</th></tr>
<tr><th colspan="2">Image Latent</th><th colspan="2">Image Explicit</th><th colspan="2">Text Latent</th><th colspan="2">Text Explicit</th></tr>
<tr><th>Layer Name</th><th>Weights</th><th>Layer Name</th><th>Weights</th><th>Layer Name</th><th>Weights</th><th>Layer Name</th><th>Weights</th></tr>
<tr><td>Input Layer</td><td>120 x 120 x 3</td><td rowspan="3">Input Layer</td><td rowspan="3">120 x 120 x 3</td><td>Input Layer</td><td>100</td><td rowspan="3">Input Layer</td><td rowspan="3">100</td></tr>
<tr><td>Conv2D</td><td>filters=32,kernel_size=(2,2)</td><td>Embedding layer</td><td>1000 x 100</td></tr>
<tr><td>Activation</td><td>Relu</td><td rowspan="2">Dropout</td><td rowspan="2">0.5</td></tr>
<tr><td>Dropout</td><td>0.8</td><td rowspan="4">Dense</td><td rowspan="4">128</td><td rowspan="4">Dense</td><td rowspan="4">128</td></tr>
<tr><td>MaxPooling2D</td><td>pool_size=(2,2)</td><td>Conv1D</td><td>lters= 10, kernel_size=</td></tr>
<tr><td>Conv2D</td><td>filters=32,kernel_size=(2,2)</td><td>MaxPooling1D</td><td>pool_size=2</td></tr>
<tr><td>Activation</td><td>Relu</td><td colspan="2">Flatten</td></tr>
<tr><td>Dropout</td><td>0.8</td><td colspan="2" rowspan="6">Batch Normalization</td><td rowspan="2">Dense</td><td rowspan="2">128</td><td colspan="2" rowspan="6">Batch Normalization</td></tr>
<tr><td>MaxPooling2D</td><td>pool_size=(2,2)</td></tr>
<tr><td>Conv2D</td><td>filters=32,kernel_size=(2,2)</td><td colspan="2" rowspan="2">Batch Normalization</td></tr>
<tr><td>Activation</td><td>Relu</td></tr>
<tr><td>Dropout</td><td>0.5</td><td rowspan="2">Activation</td><td rowspan="2">Relu</td></tr>
<tr><td>MaxPooling2D</td><td>pool_size=(2,2)</td></tr>
<tr><td colspan="2">Flatten</td><td rowspan="4">Activation</td><td rowspan="4">Relu</td><td rowspan="4">Dropout</td><td rowspan="4">0.8</td><td rowspan="4">Activation</td><td rowspan="4">Relu</td></tr>
<tr><td>Dense</td><td>128</td></tr>
<tr><td colspan="2">Batch Normalization</td></tr>
<tr><td>Activation</td><td>Relu</td></tr>
<tr><td colspan="2">Merge</td><td colspan="2">Merge</td><td colspan="2">Merge</td><td colspan="2">Merge</td></tr>
<tr><td colspan="4">Merge</td><td colspan="4">Merge</td></tr>
<tr><td colspan="8">Merge</td></tr>
<tr><td colspan="8">Activation - Relu</td></tr>
<tr><td colspan="8">Dense - 128</td></tr>
<tr><td colspan="8">Batch Normalization</td></tr>
<tr><td colspan="8">Dense - 1</td></tr>
<tr><td colspan="8">Activation - Sigmoid</td></tr>
</table>

IV. Experiment setting & Results.

Since the vast amount dataset size and keeping in mind the limited computation resources we've, the models discussed before were experimented on only 10% of the dataset extracted by a specially developed python script. We ensure that the skewness of the dataset is maintained by selecting equal number of true and fake news samples. The samples in the dataset can be divided into 2-way, 3-way and 6-way samples. For the initial stage, all the experiments were designed to classify the samples in only 2-way. Keeping into the account of limited memory, the image size is resized to (50,50,3) in some of the experiments. **Adam optimizer** with **learning rate *1e-3*** is used in each of the experiment.

The results of each model are discussed at the point of explanation of each model.

V. Comparison & Discussion

All the experiment settings and their results were displayed in the following table. Of all, 6 were unimodal (3 image + 3 text) including transfer learning algorithms and 1 Multimodal algorithm. Based on unimodal image classification, Transfer learning using ResNet-50 gives better results in classification of samples with 72.5% validation accuracy with minimal loss compared to other unimodal image models. This shows how effective a pre-trained model can extract features of an image compared to the newly trained models.

In unimodal text classification, the LSTM model gives better results in classification of test samples with the test/validation accuracy of 77% and with low loss than CNN models.

The multimodal based algorithm *TI-CNN* shows an average result with an accuracy of 75% model accuracy and 75% validation / test accuracy. This may due to the limited dataset we've provided to the model. Compared to unimodal models, the multimodal shows a good training and validation accuracy with low training and validation loss.

We believe that the model's accuracy can be eventually increased If total 100% of dataset is given to the model and fine tuning the hyper-parameters of the model.

VI. Conclusion & Future Works.

In this paper we discussed the adverse effects of fake news in the society and the importance of building machine learning models for detection of fake news. Apart from that we also discussed the importance of using multimodal (image + text) data compared to the

S.No	Model Name	Model Type	Text Input	Image Input	Epochs	Model Accuracy	Model Loss	Validation Accuracy	Validation Loss
1	**Simple CNN**	Unimodal	-	120X120x1	15	0.82	0.42	0.57	0.92
2	**Transfer Learning (Resnet 50)**	Unimodal	-	120x120x3	10	0.80	0.40	0.725	0.57
3	**CNN + Dense**	Unimodal	-	50x50x3	10	0.58	0.65	0.55	0.69
4	**CNN Based : Text**	Unimodal	100	-	100	0.63	0.60	0.58	0.60
5	**CNN +Dense: Text**	Unimodal	100	-	100	0.75	0.53	0.75	0.54
6	**LSTM : Text**	Unimodal	100	-	25	0.86	0.30	0.77	0.60
7	**TI-CNN : Multimodal**	Multimodal	100	50x50x3	50	0.75	0.50	0.75	0.50

unimodal (only text/image) in the scenarios like Fake News Detection. We analysed 6 unimodal (image + text) based models including transfer learning techniques and 1 Multimodal based on CNN and found out how effective a multimodal based neural network can perform the job compared to unimodal frameworks.

We believe, there is still room for improvement on longer length articles and more complex fusion techniques to understand how different modalities play a role in multimodal fake news detection.

In future, Multi-Modal for fake news detection can be made which can detect fake news for 3-way and 6-way with much more improvement in accuracy and loss.

VII. References.

[I]. TI-CNN: Convolutional Neural Networks for Fake News Detection [Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, Philip S. Yu]

[II]. M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein, "A stylometric inquiry into hyperpartisan and fake news," CoRR, vol. abs/1702.05638, 2017.

[III]. K. Shu, S. Wang, and H. Liu, "Exploiting tri-relationship for fake news detection," CoRR, vol. abs/1712.07709, 2017.

[IV]. J. Z. Pan, S. Pavlova, C. Li, N. Li, Y. Li, and J. Liu, "Content based fake news detection using knowledge graphs," in International Semantic Web Conference (1), ser. Lecture Notes in Computer Science, vol. 11136. Springer, 2018, pp. 669–683

[V]. Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, ser. AAAI'15. AAAI Press, 2015, pp. 2181–2187. [Online]. Available: http://dl.acm.org/citation.cfm?id=2886521.2886624

[VI]. M. Huh, A. Liu, A. Owens, and A. A. Efros, "Fighting fake news: Image splice detection via learned self-consistency," CoRR, vol. abs/1805.04096, 2018

[VII]. F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, "Detection of gan-generated fake images over social networks," in 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), April 2018, pp. 384–389.

[VIII]. Z. Jin, J. Cao, H. Guo, Y. Zhang, and J. Luo, "Multimodal fusion with recurrent neural networks for rumor detection on microblogs," in Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017, 2017, pp. 795–816. [Online]. Available: https://doi.org/10.1145/3123266.3123454

[IX]. Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, and J. 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, ser. KDD '18. New York, NY, USA: ACM, 2018, pp. 849–857. [Online]. Available: http://doi.acm.org/10.1145/3219819.3219903

[X]. D. Khattar, J. S. Goud, M. Gupta, and V. Varma, "Mvae: Multimodal variational autoencoder for fake news detection," in The World Wide Web Conference, ser. WWW '19. New York, NY, USA: ACM, 2019, pp. 2915–2921. [Online]. Available: http://doi.acm.org/10.1145/3308558.3313552

[XI]. r/Fakeddit: A New Multimodal Benchmark Dataset for Fine-grained Fake News Detection by Kai Nakamura, Sharon Levy, William Yang Wang.