International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9, Issue-7, May 2020

Detection of Fake Doctored Images used to Circulate Fake News

R. Kayalvizhi, Abhisek Bharadwaz, Monisankar Baruah

Abstract: Over the few years the world has seen a surge in fake news and some people are even calling it an epidemic. Misleading false articles are sold as news items over social media, whatsapp etc where no proper barrier is set to check the authenticity of posts. And not only articles but news items also contain images which are doctored to mislead the public or cause sabotage. Hence a proper barrier to check for authenticity of images related to news items is absolutely necessary. And hence classification of images(related to news items) on the basis of authenticity is imminent. This paper discusses the possibilities of identifying fake images using machine learning techniques. This is an introduction into fake news detection using the latest evolving neural network models.

Keywords : Machine Learning(ML), Neural Networks, Fake News, Fake Images.

I. INTRODUCTION

Computer Vision(CV) or popularly known as CV is a field of study that allows a computer to see and visualize digital media such as image and video and understand the content in the media element. Many applications such as object detection, object classification, object recognition etc uses computer vision to train the computer to recognize the object, text, person in the provided image or video.

Humans receive visual signals from the immediate surrounding and process those signals in their visual cortex. This data is vivid and large and contains various distinct objects and scenes to help identify the specific image and those identities are stored in our memory. Image recognition in computer vision does something similar by perceiving it as a set of vectors or as rasters. Raster images are sets of pixels with discrete numerical values for colors while vector images are sets of color annotated polygons as shown in figure (1).

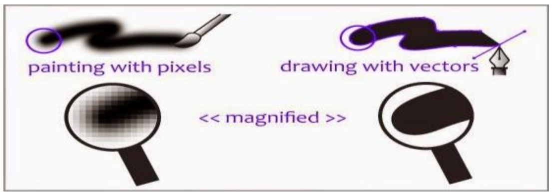

Fig 1. Pixel and vector painting

Revised Manuscript Received on May 30, 2020.
***** Correspondence Author
Dr. R. Kayalvizhi*, Assistant Professor, Department of Computer Science And Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai.
Mr. Abhisek Bharadwaz, Student**,** B.Tech, Computer Science and Engineering, SRM Institute of Science and Technology.
Mr. Monisankar Baruah , Student**,** B.Tech, Computer Science and Engineering, SRM Institute of Science and Technology.

The history of image recognition can be traced back to experiments by Hubel and Weisel in 1959, 1962 and 1968, where they observed that different cells of the brain of a cat were responsive to different types of visual stimuli. In 1980, a network architecture was first introduced by the name of Neurocognitron by Fukushima, which contained the idea of simple and complex cells by the results of Hubel and Weisel. In this model alternating layers referring to simple and complex cells were present.

The layers referring to simple cells had modifiable parameters whereas the layers referring to complex cells performed pooling. In 1998, Yann LeCun first showed the example of applying backpropagation and Gradient based learning to train Convolutional Neural Network that did quite well on document recognition.

This work was widely used in recognition of zip codes in the postal service. But beyond that it wasn't able to scale to more complex data as digits were fairly simple and were of a limited set. In 2012, the first strong results in image recognition were obtained in 2012 by Alex Kirzhevsky by the introduction of Alexnet.

Since then CNNs are widely used from self driving cars to medical imaging. Resnets and Densenets are some of the current popular evolved forms of CNNs. This paper discusses the possibilities of application of Convolutional Neural Networks in detecting doctored images used for fake news circulation.

This paper hence discusses the possibilities of detecting doctored images using Resnets and Densenets by classification of memes from news. Since to create a meme, people usually photoshop a single or multiple images, it can be said that all memes have this feature that they are all photoshopped. And also most of the doctored images used for propagating fake news also has the same feature therefore a model trained on a dataset of news and a dataset of memes basically classifies a doctored image to a non doctored one. Before going further it is necessary to discuss what Convolutional Neural Networks really are.

From the name itself the word, convolution which is a mathematical linear operation, is applied to the layers of a neural network instead of simple matrix multiplication. Each layer has a specific filter which is used for feature extraction. And these extracted features are then used to predict the images whether it matches the trained ones.

The number of layers is denoted as the depth of the network and more the number of layers in a CNN, the better it can be trained and the more efficient it can predict. But with depth comes the challenge to train.

A team of Microsoft researchers proposed a residual learning framework which eases the training process and also using this residual learning framework, relatively deeper networks can be trained.

Similarly using DenseNets(Densely Connected Networks) relatively deeper networks can be trained which uses the concept of using shorter connections in the layers close to the input and output and also the concept of connecting each layer to every other layer in a feed forward fashion.
In standard CNNs each layer receives input from it's previous layer and in the output layer generates the features. In figure (2) a plain feed forward neural network's connectivity is shown.

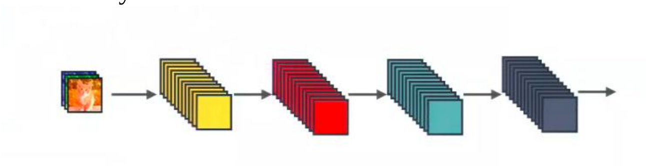

Fig 2. Plain Neural Network.

In the ResNet (Residual Networks)[13] model, the connectivity has layer skips in between as shown in figure (3). The ResNet model also shows the presence of identity mappings promoting gradient propagation.

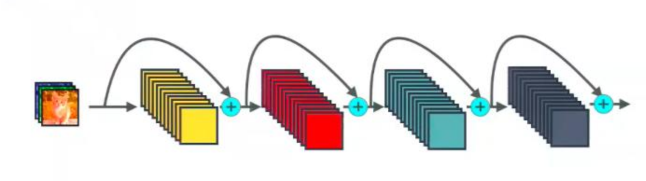

Fig 3. Connectivity in ResNet.

In DenseNets[14], the connectivity between the different layers is called dense connectivity and there may be several parallel skips in between the layers as shown in figure (4) where the layers can be skipped during the feed forward process. In this type of connectivity, every single layer receives input from all of it's preceding layers.

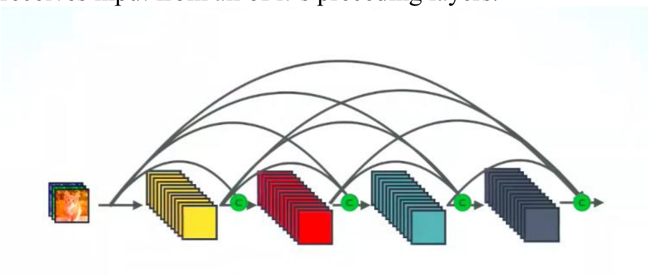

Fig 4. Connectivity in DenseNets.

In the Dense connectivity, the input from the preceding layers is combined with channel wise concatenation. This also results in better gradient flow as the error signal will flow to all the layers including the starting layer. As in traditional CNNs for N layers, there are N connections present while in Densely Connected Neural Networks (DenseNets), the number of connections for N layers is (N+1)N/2.
In this paper, two datasets, one being a dataset of memes and the other being a dataset of news items are used and trained in ResNet 50, ResNet 152, DenseNet 121 respectively.
possible.

II. LITERATURE SURVEY

Krizhevsky et al. [4] laid the foundation work for the modern day image recognition and anal-ysis. The architecture discussed in this paper consisted of eight layers, out of which five are convolutional layers and three of them are fully connected layers. It also supports the following characteristics of "Rectified Linear Unit (ReLU) non linearity", "Multiple GPUs" and overlap-ping pooling. This model won the 2012 "ImageNet Competition" with a top 5 error rate of 15.3.
He et al. [2] discusses an architecture which is more of an extension of the previously dis-cussed alexnet. At first it showed that depth always doesn't necessarily do a better job intraining and that a threshold of depth is present. It has a fundamental building block called identity where a previous layer is added to a future layer in a process called additive merging which results in better learning of the residuals.
Concatenation is the property used in DenseNets. To improve information flow between layers,direct connections from one layer to all subsequent layers are introduced. Consequently, onelayer's preceding layers send it it's respective features-maps.
The problem addressed here by Badrinarayanan et al. [1] is pixel wise segmentation of images through labelling. The Segnet architecture consists of an encoder and decoder pair which create feature maps for classification. The advantages of this architecture is that it showed improved boundary delineation and lesser number of parameters.
Here Jégou et al. [3] discusses a Fully Convolutional DenseNet used for Semantic Seg-mentation. Deep Neural Networks often have to deal with the problem of gradient vanishing.Jégou et al. [3] discusses gradient vanishing problem's effects and affects on a dense deep neural network and also discusses normalization techniques to compensate the loss.
Simonyan and Zisserman [6] discuss that CNN's which are primarily used for image detec-tion are further altered and evolved by increasing the depth. This can be done by adding very small(3x3) convolutional filters.
Wang [7]'s paper presents a publicly available dataset called "liar" which can be used to train machine learning models for detection of fake news. It consists of 128000 short statements along with links to source documents in each case.

III. IMPLEMENTATION

If To detect images correctly it is needed to train the machine learning model with a large amount of data. This leads to the first obstacle of collecting a large amount of images from the websites to train the CNN model. To tackle this, a twitter based web scraper is used. The scraper obtains the tweets from credible news accounts like BBC news, CNN (Cable News Network) news and India News and filters through the tweet to get the URL for the related news article for that tweet. After obtaining the link, the webpage is parsed into a HTML format and the image URLs are extracted from it using regular expressions. After obtaining the image URLs, the images are individually downloaded and stored to form the dataset. After scraping for images, we end up with a total of 1000 images for training and 200 images for testing and validation.

Most images that are circulated for spreading fake news are circulated as memes. It can also be said that all memes are not fake news circulating images but the vice versa can be considered true. Hence a dataset of memes was collected by the use of a reddit scrapper which scraped memes of a particular subreddits in reddit.

The training and testing data are separated into two classes ['News', 'Memes'] to segregate the data. These images are preprocessed and transformed to train the model. The Images are normalized and resized to meet the resolution (224*224) required for the densenet and resnet network. The data is also rotated and center cropped to remove unnecessary noise from the images.

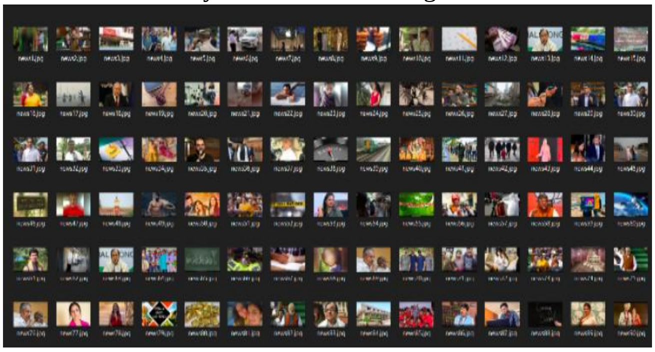

Fig 4. The dataset

Figure (4) shows the dataset where 1000 images of news related images and 1000 images of comics or memes are taken as training data and with a batch size of 64 while the test data consists of 200 images of the same classes with a batch size of 32 for a smaller pool of data.

A pretrained model of ResNet-152, ResNet-50 and DenseNet-121 are used so that it contains weights and features of the similar dataset that can be transferred to train the given dataset without using much computational resource and time. At the end of each CNN model, we add our own linear layer with a Rectified Linear activation function followed by a dropout layer and the outputs are passed through a LogSoftmax function to plot the output in a range of [0-1].

$$\sigma(z_i) = e^{z_i} / \sum_{j=1}^{K} e^{z_j} \quad \text{for } i = 1..K \text{ and } z = (z_1..z_K) \in R^K \quad (1)$$

Equation (1) shows the softmax function where the standard exponential function is applied to all input vectors z for i inputs upto K and this is divided by the sum of all exponential input vectors to normalize the output so that the sum of the components equate to 1.

$$L(y) = \log(\sigma(z_i)) \quad (2)$$

Equation (2) derives from equation (1) where the Log of the softmax function is used to get the output. This improves the numerical performance of the code and optimizes the gradients. The log softmax function heavily penalizes the network if the output is incorrect hence updating the weights and biases more aggressively.

The ResNet model gives 2048 outputs from the convolutional layers and those outputs are passed through the Linear layers while the DenseNet model gives 1024 outputs from it's convolutional to be passed through the Linear layer. All the models are given 3 Linear layers and the first two layers are followed by a ReLU[15] activation function and a dropout layer with a dropout probability of 20%. The last Linear layer is followed by a LogSoftmax activation function to plot the predictions between [0-1]. The ResNet has 3 linear layers following the convolutional layers and in-between these layers are ReLU activation functions and a dropout layer of probability 0.2 to turn off random nodes during the training process so as to not overfit the model. The first linear layer has 2048 inputs from the convolutional layers and the 2048 inputs are forwarded to 1024 nodes in the next linear layer. Before the next linear layer, the input passes through the ReLU activation function and a dropout layer to normalize the input for the next layer. Similarly the next linear layer has 1024 inputs and outputs 256 inputs for the next layer which is again followed by a ReLU activation function and a dropout layer. The last linear layer takes in 256 inputs and gives 2 outputs for the two classes in which the dataset is arranged. The final output is passed through a log softmax function where the output is normalized and the wrong predictions are heavily penalized.

Similarly for the DenseNet model, 3 linear layers are added with ReLU activation functions and dropout layers in between those layers which is followed by the log softmax function to normalize the final output. The first linear layer takes in 1024 inputs from the convolutional layers and produces 512 outputs for the next layer. This is followed by the ReLU activation function and a dropout layer with a probability of 0.2. The next linear layer takes in 512 inputs and produces 256 outputs for the next linear layer followed by a similar ReLU activation function and a dropout layer. The last linear layer takes in 256 inputs and produces 2 outputs for the 2 classes in the data set and these outputs are passed through the log softmax function to normalize the output and penalize the model heavily for wrong outputs.

The ReLU (Rectified Linear Unit) activation function used takes the input from the layer and outputs the input if the input is greater than 0 or outputs 0 if the input is less than 0. The ReLU activation function is used in this paper because it overcomes the vanishing gradient problem faced when training a network. The vanishing gradient problem is the one where a deep neural network or a recurrent neural network cannot produce useful gradient information from the output of the layers to feed to the next layer as input. This is solved by using the ReLU activation function which is used in this paper. The ReLU activation function also allows models to train faster and better because of the simplicity of the activation function.

The ResNet or Residual Neural Network used in this paper is based on the construct of the pyramidal neurons in the cerebral cortex region of the brain where the network can skip layers and has ReLU activation and batch normalization in between layers. For a single skip, the layers may be numbered as L-2 to L or L+2 to L. The two forms of indexing are convenient to describe a forward skip or a backward skip. As a signal flows forward through the network it is easier to describe the skip as L+K from a given layer, but as a learning rule (back propagation) it is easier to describe which activation layer you reuse as L-K, where K-1 is the skip number. The ResNet-152 is a 152 layer network consisting of CNN layers and Linear layers while the ResNet-50 consists of 50 layers.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig 5. ResNet architecture

Figure (5) shows the ResNet-152 architecture which has 1 convolutional layer with a (7x7) kernel and stride 2, a pooling layer of filter size (3x3) and stride 2, a 2nd convolutional layer with 3 (3x3) kernels, a 3rd convolutional layer with 8 (3x3) kernels, a 4th convolutional layer with 36 (3x3) kernels and a final convolutional layer with 3 (3x3) kernels and it outputs a feature set containing 2048 outputs. This is passed on to the Linear layer which is given above and the output is passed through the log softmax activation function. The ResNet-50 has a similar architecture but with 4 (3x3) kernels in the 3rd convolutional layer and 6 (3x) kernels in the 4th layer. This also outputs a feature set containing 2048 outputs.

The DensNet or Densely Connected Neural Networks are similar to ResNets with the exceptions that it does several parallel skips of the layers during training.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv; 3 × 3 conv] × 24	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48	[1 × 1 conv; 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv; 3 × 3 conv] × 16	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Fig 6. DenseNet Architecture

Figure (6) shows DenseNets with different number of layers and in this paper DenseNet-121 is used which has 121 layers consisting of 4 dense blocks in which the first dense block has 6 (1x1) and (3x3) convolutional layer, the second dense block has 12 (1x1) and (3x3) convolutional layer, the 3rd dense block has 24 (1x1) and (3x3) convolutional layer and the final dense block has 16 (1x1) and (3x3) convolutional layer. The output contains a feature set of 1024 outputs which is passed through the linear layer given above and the log softmax function is used on the output to normalize the result. The first layer the input passes through has a convolutional layer of kernel size (7x7) with stride 2 and it is followed by a Max Pooling layer of kernel size (3x3) with stride 2. In between the dense blocks (1x1) convolutional layers and followed by average pool layers of kernel size (2x2).

For, computing the loss, the NLLLoss (Negative Log Likelihood Loss) function is used which is presented by:

$$\square(\square) = -\square\square\square(\square). \tag{3}$$

In equation (3) the negative log of the input y is calculated to give L(y)where y is the has to be a tensor of size either (minibatch ,C) or (minibatch, C, d1, d2,...,dk) with K1 for K-dimensional cases and the target or the output will be a value N where each value is 0output[i]C-1 or N, d1, d2,...,dk with K1 in case of K dimensional outputs. The log probabilities can be obtained by adding the Log Softmax function (2) to the model. The loss is calculated as in equation (4) and (5) .

$$\square\square\square\square(\square, \square\square\square\square\square) = -\square\square\square\left(\frac{\square\square\square(\square[\square\square\square\square\square])}{\sum_{\square} \square\square\square(\square[\square])}\right) \tag{4}$$

$$\square\square\square\square(\square, \square\square\square\square\square) = -\square[\square\square\square\square\square] + \square\square\square\left(\sum_{\square} \square\square\square(\square[\square])\right) \tag{5}$$

The loss function takes in the log probabilities and the class and it is equated by taking the negative log of the difference between exponential probability of the class item and the sum of all the exponential class scores as shown in equation (4) and can also be written as summation of the negative class probability and the log of the summation of the exponential probabilities of the class item as shown in equation (5).

For optimizing the weights and biases, the Adam[12] optimizer is used which is used to update the weights based on the computed gradients. It takes in the model parameters and the learning rate which is set to 0.02 and this signifies how much of the weights are changed in each iteration to not overfit the model.

All the three models, DenseNet-121, ResNet-50 and ResNet-152 are trained for 30 iterations since the models do not improve accuracy beyond that and the loss also increases after 30 iterations. The models with the highest accuracies are saved and used for testing and predicting the image and ResNet-152 is used for testing and deployment.

However, in case the image contains any kind of textual features the text is extracted by using pytesseract which is a python wrapper for Google's Tesseract-OCR[16] engine which is an Optical Character Recognition (OCR) tool used to extract text embedded into the image. The text of that image can be extracted and it's content can be analyzed using NLP techniques. It can be fact checked and be predicted whether the text in the given image is authentic or not. For the purpose of fact checking Stance Detection as discussed by Benjamin Riedel[18] can be used. For the purpose of fact checking, in a nutshell, Stance Detection analyzes the input content on the basis of what the other reputable resources on the web have to say about it. Stance detection has three main stancs namely Agree, Discuss, Unrelated and Disagree.

These four stances denote whether a heading agrees with a particular paragraph, or that the body and the headline are unrelated,that it is discussable, or that the body disagrees with the headline respectively. One dataset contains the body of the text with a respective body ID. The other dataset contains a body element and with it it's stance. So with the help of these two datasets together a machine learning model can be trained which recognizes if a certain paragraph or a text content about a headline, either agrees with the headline, gives a neutral comment or does it disagree with the headline. The final output of the model determines whether the given text given is true or false.

The models are deployed in a browser where the person inputs an image file and the browser gives the result whether the image itself is a news and also, if the text is given, checks for the legitimacy of the text.

IV. RESULT AND DISCUSSION

The ResNet-152 model was used for the final detection because it had an accuracy of over 90 percent on the testing dataset and hence gave optimal outputs upon entering various images.

Upon inputting an image of Donald Trump as shown in figure (7), we can find that the picture is classified as a meme because of the various image adaptations of Donald Trump being used to make parodies of him while the textual data confirms it's legitimacy.

Fig 7. Donald Trump on Hussein.

The result page shows the prediction made by the image prediction model and the text prediction model.

Fig 8. The result

Figure (8) shows a demo of the final result and outputs the following : "This image appears to be a meme".

V. CONCLUSION

This paper was a study to detect doctored images relevant as a news item from the real ones. This paper tries to find out relevant ties between various properly doctored images relevant to news items through a modern evolved neural network which can handle classification problems with relatively higher features.

REFERENCES

1. Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015). "Segnet: A deep convolutional encoder-decoder architecture for image segmentation.
2. He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Deep residual learning for image recogni-tion.
3. Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., and Bengio, Y. (2017). "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation."2017IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 1175–1183.
4. Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks."Proceedings of the 25th International Conference onNeural Information Processing Systems - Volume 1, NIPS'12, Red Hook, NY, USA. CurranAssociates Inc., 1097–1105.
5. Riedel, B., Augenstein, I., Spithourakis, G. P., and Riedel, S. (2017). "A simple but tough-to-beat baseline for the fake news challenge stance detection task.
6. Simonyan, K. and Zisserman, A. (2014). "Very deep convolutional networks for large-scale image recognition.
7. Wang, W. Y. (2017). ""liar, liar pants on fire": A new benchmark dataset for fake news detection."CoRR, abs/1705.00648.
8. Brostow, Gabriel J. et al. "Segmentation and Recognition Using Structure from Motion Point Clouds." ECCV (2008).
9. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2018 Apr;40(4):834-848. DOI: 10.1109/tpami.2017.2699184
10. Deng, Jia & Dong, Wei & Socher, Richard & Li, Li-Jia & Li, Kai & Li, Fei Fei. (2009). ImageNet: a Large-Scale Hierarchical Image Database. IEEE Conference on Computer Vision and Pattern Recognition. 248-255. 10.1109/CVPR.2009.5206848.
11. Gatta, Carlo et al. "Unrolling Loopy Top-Down Semantic Feedback in Convolutional Deep Networks." 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (2014): 504-511.
12. Drozdzal, Michal et al. "The Importance of Skip Connections in Biomedical Image Segmentation." ArXiv abs/1608.04117 (2016): n. pag.

AUTHORS PROFILE

Dr. R. Kayalvizhi works as an Assistant Professor in the Department of Computer Science And Engineering in SRM Institute of Science and Technology, Kattankulathur, Chennai. Dr Kayalvizhi has obtained her B.E in Computer Science and Engineering from Madras University in the year 2000, her M.E in Embedded System Technologies from CEG Campus Anna University in 2007 and Ph.D in Wireless Sensor Security from MIT Campus Anna University in 2016. Dr. R. Kayalvizhi's area is in Wireless Sensor Security, Real Time Embedded System. Some of Dr Kayalvizhi's publications include "EOG based hand controlled glove designed using arduino", Journal of International Pharmaceutical Research, 46(4): pp 80-85, 2019. (Scopus); "A Survey on Online Click Fraud Execution and Analysis", International Journal of Applied Engineering Research, 2018. (Scopus).

Mr. Abhisek Bharadwaz is a final year undergraduate student currently pursuing his Bachelor of Technology(B.Tech)(2016-2020) in Computer Science and Engineering from SRM Institute of Science and Technology. His areas of Interests include Machine Learning and Artificial Intelligence, Game Development, Computer Hardware and IoT(Internet of Things) . Mr Abhisek is an aspiring researcher in the field of Artificial Intelligence,Machine Learning, Image Processing and Computer Vision and the above paper is his first step towards his aspiration. Mr. Abhisek has worked on personal projects like ultrasonic RADAR System, Animal Classifier using CNNs, Style transfer using RNNs, Project Boost and Air Assault on Unity3D and group projects like Health Konnect and Blind Walking Stick based on IoT.

He has previously worked as an Trainee in BSNL ,Ghaziabad and also as an Intern in the ICE Department of ONGC, Scope Minar, Delhi and worked on MPR Data Management.

Mr. Monisankar Baruah is a final year undergraduate student currently pursuing his Bachelor of Technology(B.Tech)(2016-2020) in Computer Science and Engineering from SRM Institute of Science and Technology. His areas of Interests include Machine Learning, Artificial Intelligence and Natural Language Processing(NLP). Mr Monisankar is an aspiring researcher in the fields of Artificial Intelligence,Machine Learning, Image Processing and Computer Vision and the above paper is his first step towards his aspiration. Mr Monisankar has previously worked on personal projects like Credit Card fraud Detection, Stock Market Clustering and Python Motion detector. He has previously worked as an Intern in Kodekata Technologies Guwahati and also as an Intern in the IT department of NTPC Ltd, Bongaigaon.