Image Forgery Detection

Mansi Patel[1], Megha Mathpal[2]

#Sacramento State University, California
United States of America

[1]mansipatel@csus.edu

[2]meghamathpal@csus.edu

***Abstract*— Currently, digital images and videos have high importance because they have become the main carriers of information. However, the relative ease of tampering with images and videos makes their authenticity not trustful. In this paper we present image forgery detection methods based on deep learning techniques which uses Convolution neural network (CNN) and Transfer Learning to automatically learn that the image is pristine (never manipulated) or fake from the given input of images.**

I. Introduction

Powerful image editing tools are very common and easy to use these days. With the help of such tools some forgeries can be done by adding or removing some information on the digital images. To detect these types of forgeries such as region duplication, the importance of forgery detection is much increased. Copy-move forgery and image splicing are most commonly used methods. In copy-move forgery it copies a part of the image and pastes it into another part of the image in that same image while in image splicing two different images are used to copy content from one and past it to another image [1]. These types of images are prime sources of fake news and are often used in malevolent ways. Image originality and authenticity has become important in many real time applications like banking, news, legal processing documents, crime investigation, scientific processes etc. Because of such image forgery it may result into major security threat as any end user can tamer or modify the visual contents of original image without keeping any visibly known traces. Even the eye of a highly competent forensic expert can miss certain signs of a fake, potentially allowing forged images to be accepted as court evidence. Before action can be taken on basis of a questionable image, we must verify its authenticity. The IEEE Information Forensics and Security Technical Committee (IFS-TC) launched a detection and localization forensics challenge, the First Image Forensics Challenge in 2013 to solve this problem [2]. The image data set is been provided with set of forged and pristine images and we try to solve this problem by using CNN which is a class of deep neural network to identify the image is fake or pristine by binary classification and then use Transfer Learning to improve the performance of our model. In this paper we will first understand the data that is been provided and perform, data cleaning, and data pre-processing on it. Detail study of CNN architecture and Transfer Learning models will be done. At the end we will evaluate the results obtained by the models and compare them.

II. Problem Formulation

The data set is obtained from Image Forensics Challenge (IFC) website [2]. They provide an open dataset of digital images it has two directories in it. One with pristine images which are original images taken under different lighting conditions with no manipulation done on it and second is forged images which are manipulated by different algorithm like clone-stamp, alpha matting etc. In total we have 1950 images in which 1050 are pristine images and 900 are fake. The set of fake images contains mask images with it. The mask images are the image portion used to hide or manipulate the original image content. Mask images have zero-pixel intensity value in some portion and non-zero in others. Mask of a fake image is a black and white (not grayscale) image describing the spliced area of the fake image. The black pixels in the mask represent the area where manipulation was performed in the source image to get the forged image, specifically it represents the spliced region. In our fake image dataset, we have 450 mask images and 450 fake images which are been manipulated by the help of mask images. We use binary classification to predict whether the given set of images falls under fake or pristine category.

III. System Design

A. System Architecture

We experimented by working on dataset in two different ways in first approach we tried to predict about the image without feature extraction by simply using CNN model and transfer learning on our image dataset. In second approach, we did feature extraction after data cleaning and then used that data samples that we obtained after feature extraction in our CNN architecture and further perform transfer learning. The first step is data cleaning and data pre-processing from which we get a noise free final dataset on which we are going to perform further actions. Once our data is clean we do feature extraction for our data and work on image at pixel level to get a better output. We applied a custom CNN architecture on that dataset and then did transfer learning to improve our results.

B. Data cleaning and pre-processing

As mentioned our data set has two directories fake and pristine having 1950 images in total, we load all the images and obtain the number of images present in each directory.

The images are of 1,3,4 channel so we understand the data by data pre-processing. The numbers which are used to specify the colour of each pixel is the number of channels each pixel has. One channel image are grayscale images, while three channel images are RGB images and four channel images stand for ARGB. In our dataset we have fake images with 3,4 channel Pristine and mask images have 1,3,4 channels. Next, we do is data cleaning for our dataset for pristine images we have only few data for one and 4 channel image which gives a hint that these images can be just noise. So, we remove those images and work only on 3 channel images for pristine images. After data cleaning for pristine we got total 1025 images left. For fake images we reshape all images to 3 channel images. Once our data cleaning is done we split our data for train and test and further implement models on it. Fig 1 shows the number of 1, 3 and 4 channel image data set for pristine, fake and mask images.

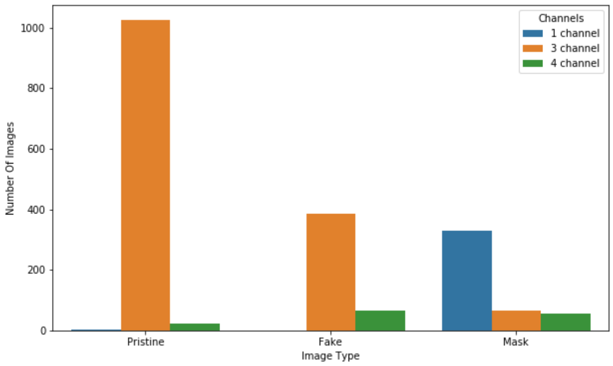


Figure 1: Data Visualization with the channel depth for fake, pristine and mask

C. *CNN with no feature extraction*

Convolutional neural networks are deep artificial neural networks that are used primarily to classify images, cluster them by similarity, and perform object recognition within scenes. We used the model same as the one is custom CNN architecture that is explained later but just without the feature extraction to compare its result to the one with the feature extraction. In this we use image data generator that Generate batches of tensor image data with real-time data augmentation, the data will be looped over (in batches). We also use fit generator that trains the model on data generated batch-by-batch by a Python generator that is run in parallel to the model for efficiency. [5]

D. *CNN with Feature extraction*

To detect the image forgery, we must work on the data set at pixel level, so we must transform our dataset into such state we used methodology mentioned in one research paper [3]. For every fake image, we have a corresponding mask. We use that mask to sample the fake image along the boundary of the spliced region in such a way to ensure at least a 25% contribution from both forged part and unforged part of the image. These samples will have the distinguishing boundaries that would be present only in fake images. These boundaries are to be learned by the CNN we design. Since all 3 channels of the mask contain the same information (fake part of an image in different pixels), we need only 1 channel to extract samples. Fig 2 shows an example of fake image and its corresponding mask.

The grey scale images are transformed to binary by using the Otsu's threshold after we denoise our data using a Gaussian Filter. After this we did sampling of our data in which there is a moving 64 x 64 window with 8 strides in it. This sliding window moves through the fake image and counts black pixel value as 0 and checks the corresponding mask and sampling to see whether the value lies in certain interval. In Fig 3 we can see that the shape of fake and mask images is different, so we resize them all.

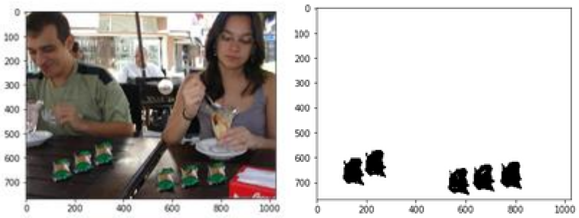
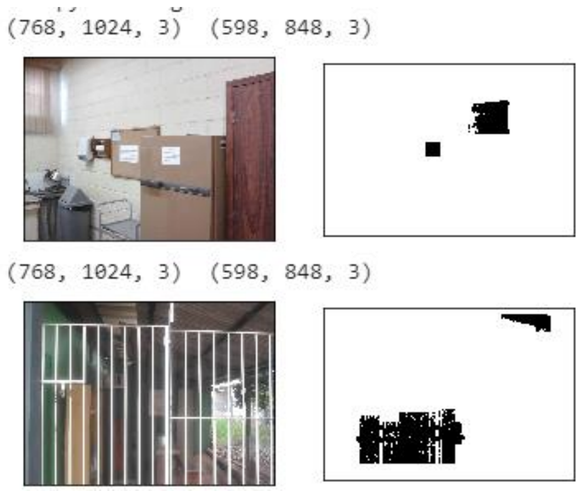


Figure 2: Example of fake image and its corresponding mask



```
(768, 1024, 3)  (598, 848, 3)
```

```
(768, 1024, 3)  (598, 848, 3)
```

Figure 3: Example of different shape for fake and mask images

After sampling our data, we got 185459 64 x 64 patches form fake image. Similarly, we created samples for pristine images as well which are 0 labelled patches. As a result, we had 185320 patches which we split as train and cross-validation. By doing so we try not to make our data overfit. As a result, we have a larger dataset of good quality input images. But the resulting dataset were huge in, so we chopped the train images to 50,000 images and 15,000 images for test.

E. Customized CNN architecture

A novel image forgery detection approach which can automatically learn feature representation based on deep learning framework mentioned in research paper, had a larger network and 128 X 128 X 3 input image size [4]. The architecture of the CNN model is shown in fig 4 is inspired from there which has 64 X 64 x 3 as input size as we have smaller network, we used half the spatial size. In Fig 4 we see the green layers are convolution layer and the blue one is max pooling. The network parameters were comparatively less to the trained models, so we did very less dropout of 0.2 to flatten the output by 20 units. Adam optimizer is used with a default value of learning rate and beta_1 and beta_2. The network was trained on sample dataset

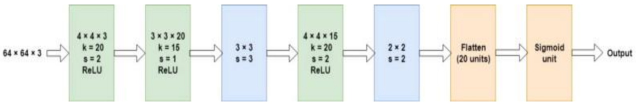


Figure 4: The CNN model architecture (used for both with feature extraction and without feature extraction)

F. Transfer Learning

Currently, transfer learning is getting very popular in the field of deep learning as it enables you to train Deep Neural Networks with comparatively little data. It is a ‘design methodology’ within Machine Learning mostly used in Computer Vision and Natural Language Processing Tasks like Sentiment Analysis, because of the huge amount of computational power that is needed for them [6]. It focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. We used VGG16 network which is trained on image dataset to vectorize the images [7]. We used bottleneck features that are output by VGG16 and build a shallow network on top of that. The top layer architecture is shown in Fig. 5 we trained our model by Adam optimizer. The input to the flattening layer is bottleneck features output by VGG16. These are tensors of shape (2×2×512) since we have used 64×64 input images.

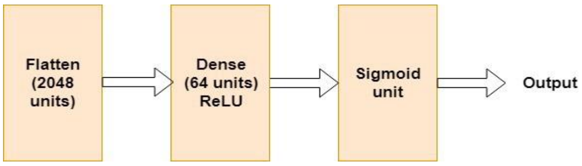


Figure 5: Top layer architecture for VGG16 (used for both with feature extraction and without feature extraction)

G. Transfer Learning with Feature extraction

Similar approach as CNN was taken with the model Transfer Learning. The prepared dataset with the binary samples of the size 64 x 64 x 3 as an input and predicting on that dataset. The model architecture for both was same, without feature extraction and with feature extraction to compare accuracies.

IV. Experimental Evaluation

A. Methodology

After understanding the data, for, working with the 3-channel depth for fake and pristine images while masks were used in binary. We decided to work with 2 approaches. First without feature extraction and second with feature extraction on the data. For both approaches we took the train and test data sampled into similar numbers for both classes, so we have fair prediction to the data given. The final data shape for without feature extraction contains 900 images among them 450 fake images and 450 pristine images while with feature extraction 65,000 samples which contains both pristine and fake images in relatively similar ratio.

After train-test split into 70%-30% respectively for both approaches. For the second approach we did feature extraction on our data got 25052 samples for pristine and 24948 samples for fake images which is the subset of the data samples we achieved. The experimental result for both the approaches and comparisons are shown in the result section.

The models taken here to predict are CNN architecture and Transfer Learning and comparing both models for both approaches to observe the experimental results where the models used were the same in both approaches.

We checked F1 score for all models and ROC curve for both the approaches are used to compare different methods. At the end we showed images as examples after we predicted image data.

B. Results

- Fig 6 shows the comparison between CNN model and Transfer Learning with F1 score for the one without feature extraction.
- In Fig 7 shows visual representation of data we got from confusion matrix in which we got F1 score for fake and pristine images for both CNN and transfer learning without feature extraction.
- Similarly, the comparison is shown in Fig 8 and Fig 9 for the model with feature extraction.
- Roc curve for transfer learning with feature extraction is shown in Fig 10.
- Fig 11 below shows the comparison of F1 score for model with and without feature extraction. We can clearly see the model with feature extraction gives better accuracy.

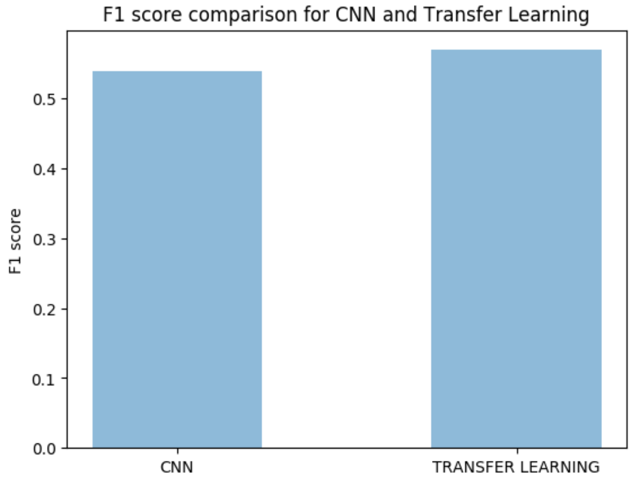


Figure 6: F1 score comparison of CNN and Transfer Learning models without the feature extraction on images

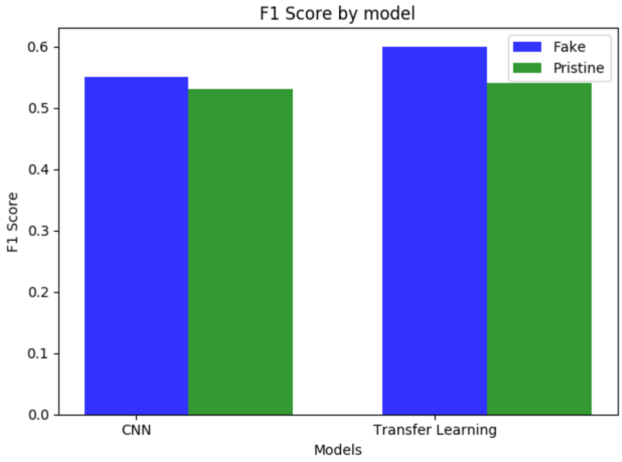


Figure 7: F1 score comparison of CNN and Transfer Learning models without the feature extraction on images for both the classes fake and pristine

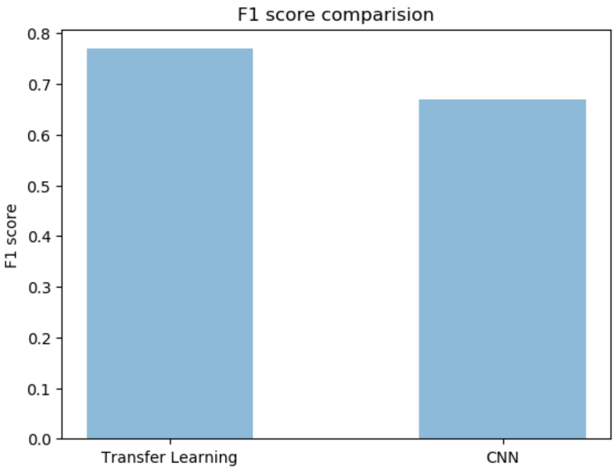


Figure 8: F1 score comparison of CNN and Transfer Learning models with the feature extraction on images

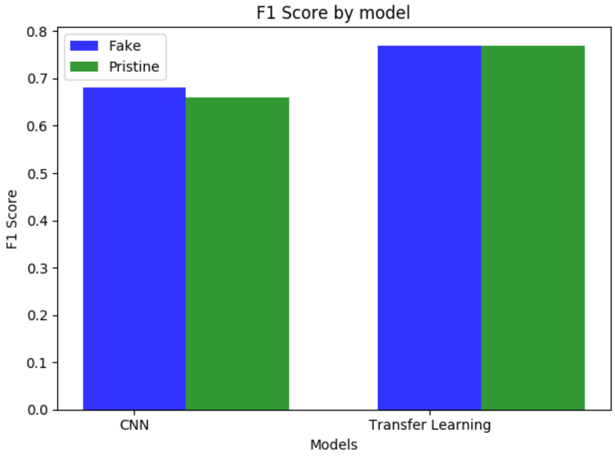


Figure 9: F1 score comparison of CNN and Transfer Learning models with the feature extraction on images

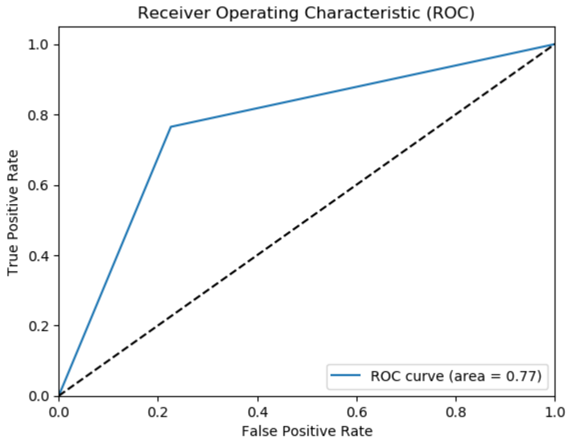


Figure 10: ROC curve for transfer learning model with feature extraction.

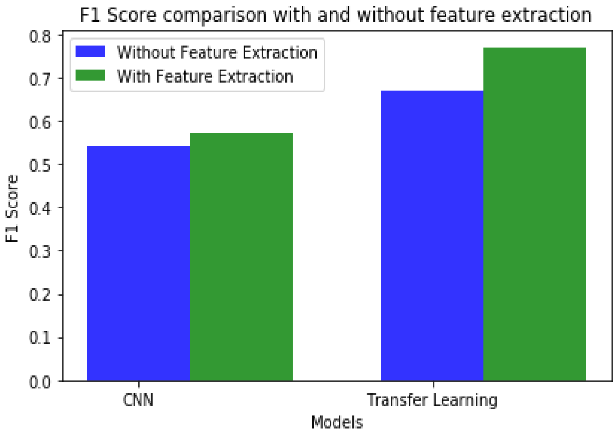


Figure 11: F1 score comparison with and without feature extraction for CNN and Transfer Models

V. Related Work

We worked on Customized CNN model based on a research paper, but they had bigger network, so we modified our CNN accordingly that was explained in one article [4]. We completely understood that and tried to implement in similar way. Unfortunately, because of low resources the image sampling was really time consuming for us, so we used just a portion of it in the models. But we were still able to get a decent accuracy score. We did feature extraction by the same idea, but we did comparison between the data set with feature extraction and without feature extraction for CNN and VGG16 models.

VI. Conclusions

We experimented with two approaches to implement the CNN and transfer learning one with and other without feature extraction. After comparing all the results, we conclude that for the first approach the model design was same, and we got better results for transfer learning than CNN. For the second approach we did feature extraction and the models work on the new data set and the results are better than the one without feature extraction moreover transfer learning still gave better accuracy for this approach as well. Fig.11 shows summary of all these experiments done.

VII. Work Division

Mansi Patel:
Data pre-processing
All Model implementation with Feature extraction
Model evaluation and comparison

Megha Mathpal:
Data Visualization
All Model implementation without feature extraction
Model evaluation and comparison

VIII. Learning Experience

It is always a new experience, working on a new dataset in Deep Learning. It was interesting working with image data set this time, here are few things:

- Learned how to handle large image dataset.
- Understanding the conversion of image data to grayscale and then binary format.
- Got better understanding of fit generator and Image data generator.
- Learned how to perform feature extraction on image dataset and understanding concepts like sliding window.
- Understanding how to make sample of images. It was too time consuming to do sampling.
- Got to know how to save large trained model into json file and easily access them later.
- Deep understanding of transfer learning
- Implementation of complex CNN model
- Learned more about neural networks and how convolution layers and max pooling, flattening etc. play an important role in getting better accuracy.
- Understanding of VGG16 network model
- Decreasing the strides and increasing the kernels to make samples from the original image results into large number of samples. This is benefited when the dataset is relatively small, by making small samples from the dataset the performance might be enhanced.

References

[1] Mohanad Fadhil Jwaid and Trupti N. Baraskar, “Study and analysis of copy-move & splicing image forgery detection techniques” in IEEE International Conference on I-SMAC 2017 pp. 697-701

[2] The Image Forensics Challenge website [online]
Available: http://ifc.recod.ic.unicamp.br/fc.website/index.py?sec=0

[3] Y. Rao and J. Ni, “A deep learning approach to detection of splicing and copy-move forgeries in images,” in Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), 2016.

[4] Vishal Singh. (2017) homepage on Towards Science. [Online]. Available: https://towardsdatascience.com/@vishu160196

[5] The Keras Blog website [online].
Available: https://blog.keras.io/index.html

[6] Niklas Donges (2017) homepage on Towards Science [Online]
Available: https://towardsdatascience.com/@n.donges

[7] Neurohive popular network VGG16-CNN for classification and detection. [online].
Available: https://neurohive.io/en/popular-networks/vgg16/