Image Forgery Detection

Anmol Jain (MT19005) and Bhakti Batra (MT19115)

I. Problem statement and motivation

The development in the field of digital images, lead to several approachable image manipulation techniques, which has made image forgery more easier. As a result it has become a keen concern for cyber crime community and if not tackled accordingly serious issues can emerge. Deep learning and computer vision community had proposed several approaches to solve this issue[1]

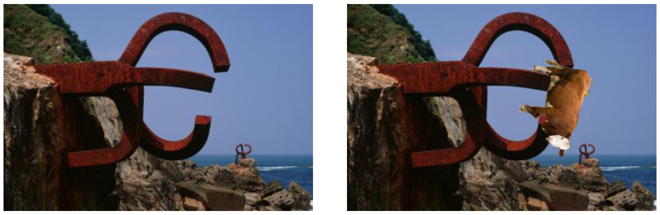
In this project we tried to solve the problem of copy move Image forgery detection, in which we have trained a deep learning model for feature engineering. For the classification purpose we have used standard classifiers and compared the performance of each of the classifier.

II. Literature review

The development in the field of digital images, lead to several approachable image manipulation techniques, which has made image forgery more easier. As a result it has become a keen concern for cyber crime community and if not tackled accordingly serious issues can emerge. Deep learning and computer vision community had proposed several approaches to solve this issue[1]

Image forgery is defined as manipulating the image in different ways. Following are the different types of manipulations [2]:

- Copy-move: a specific region from the image is copy pasted within the same image.
- Splicing: a region from an authentic image is copied into a different image.
- Removal: an image region is removed and the removed part is then in-painted.



(a) Authentic

(b) Tampered

Fig. 1: Example of tampered image (copy-move) and authentic image extracted from our data set

With the advancement of technology, image manipulations can be done so professionally that it becomes difficult for a human being to differentiate between original and forged image. An example has been shown in the Figure-1. We have targeted these kind of images in our project. Our aim is to make the machine perform better in classifying such images. In a nutshell, the idea behind copy move detection can be to detect the tampered edges which are not finely merged with the original image.

Stating about background, there have been many methods developed till now for image forgery detection which uses the computer vision libraries for feature extraction. There are also papers on efficiently extracting tampered image regions with the help of Neural Networks. In this project, instead of using the traditional approaches for feature extraction we have implemented a CNN model which is trained on sampled patches extracted from images. For the mask extraction step, the model was trained by extracting the masks manually as well using the already available ground truths of the used data set. Therefore, the project can be divided in following steps:

1) Mask extraction: Making the data set accordingly, storing the tampered and their corresponding authentic images.
2) Extracting the tampered regions patches from the manipulated images available in the training data set for training purpose.
3) Training the CNN model on the extracted patches.
4) Feature Representation and Fusion.
5) Classification and Performance Comparison.

III. Dataset Details

We have used the CASIA 2.0 dataset. It has tampered images: 5123, authentic images : 7491 and the ground truth containing the tamoered masks. For the first attempt we have not used the available ground truth and extracted it on our own. For training purpose, we have sampled 3000 tampered images and their corresponding authentic images which came out to be 1784. In order to make a balanced dataset,we randomly picked 3500 authentic images. Therefore in total we had 3000 tampered images and 5284 authentic images

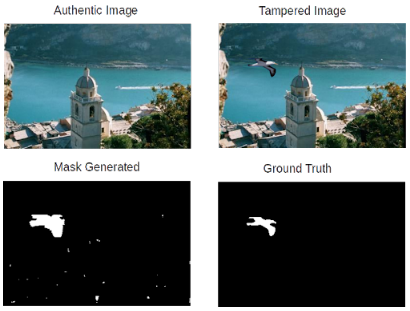


Fig. 2: Manually extracted mask and ground truth mask

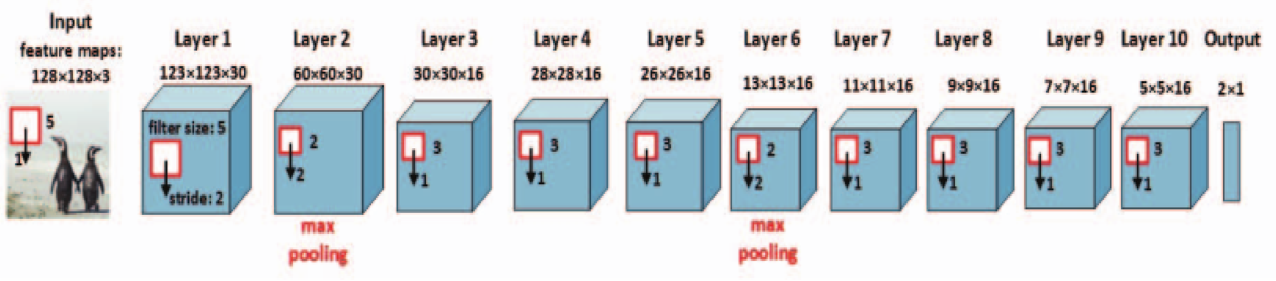


Fig. 3: CNN Architecture

in the training data set. For testing dataset, we had 1122 authentic images and 2207 tampered images.

Reason behind choosing the CASIA 2.0 dataset is that the tampering is difficult to detect in this data set. Therefore, the CNN trained for feature engineering on this data can perform better for other data sets as well.

IV. Proposed architecture

The similar CNN architecture is implemented as proposed here [3]. Additions has been done in the architecture as more SRM filters are added to the second convolution layer. This architecture was tested on CASIA-1 dataset which has easy to detect spliced images. Using the similar architecture we aim at achieving the similar accuracy for CASIA-2 dataset which has comparatively difficult to recognise forgery. Further, the accuracy has also been checked on unseen dataset as well, CASIA-1 for this project.

A. *CNN architecture*

Figure 3 [3] shows the layers of CNN model. An RGB image has been fed to the model, so the first convolution layer tend to have 3 layers with kernel size 5X5 and stride = 1. The image that has been fed is of size 128X128. Since each image of different dimension so we trained the model on the patches extracted by sliding a window of size 128X128. It has 7 conv2D layers, 2 maxpool layers and 1 fully connected layer. A softmax layer has been used for training purpose. The first two layers have kernel size of 5X5 and filters 3 and 30 respectively. The further convolution layers have filters of 16 and kernel size of 3. Finally, the dense layer has 400 layers. Therefore, given the input as a patch, it will produce an output of 400-D. Each layer uses 'relu' non linear activation. If we denote Fn(X) the feature map in the convolution layer n, with the kernel and bias defined by Wn and Bn respectively, the convolutional layer can be computed using the following formula [3]:

$$Fn(X) = pooling(fn(Fn1(X)Wn + Bn))$$

where $F0(X) = X$ the input data, $fn(\Delta)$ a non-linear activation function applied to each element of its input and $pooling(\Delta)$ the pooling operation which reduces the dimensions of the data via a max or mean operation.

B. *Patch Extraction*

In order to train the CNN for feature extraction, we have employed 2 techniques.Training the model by extracting 2 patches from each of the tampered and authentic images with the patch size = 128X128X3 where the patches extracted from tampered images must contain the forged region. In the first technique, it has been done manually but the performance decreased because we encountered noise while extracting the tampered region. Hence, model didn't get train efficiently. Therefore, we moved on to another technique where we used the available ground truth masks from here. Figure 2 shows the difference between manually extracted masks and the ground truth mask. We can extract the patches from tampered regions only. Since the patches can be large in number and we have a heavy model so we have trained it on the 2 randomly chosen patches as well on 7 randomly chosen patches.

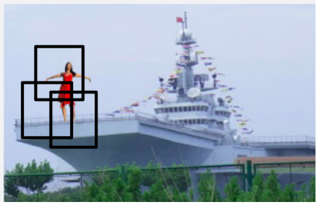


Fig. 4: Training Patches

C. *SRM Filters*

Each layer of CNN has been given weights by Xavier initialisation. In order to improve the performance of the model, the second layer has been initialised by the 30 low pass and high pass filters. These filters were introduced in [4]. SRM quantifies and truncates the output of these filters and extracts the nearby co-occurrence information as the final features. The feature obtained from this process can be regarded as a local noise descriptor [4]. The following image shows two of the 30 filters applied. Out of the 30 filters, 3 are SRM and 27 filters are low pass filters used for detecting edges. Each filter is of shape 5X5. This layer outputs the 30 channels of 124X124 dimensions.

D. *Feature Fusion*

Feature fusion is performed when all the patches of an image are represented by 400D features. These features are

$$\frac{1}{4}\begin{bmatrix}0&0&0&0&0\\0&-1&2&-1&0\\0&2&-4&2&0\\0&-1&2&-1&0\\0&0&0&0&0\end{bmatrix}\quad\frac{1}{12}\begin{bmatrix}-1&2&-2&2&-1\\2&-6&8&-6&2\\-2&8&-12&8&-2\\2&-6&8&-6&2\\-1&2&-2&2&-1\end{bmatrix}\quad\frac{1}{2}\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&1&-2&1&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}$$

Fig. 5

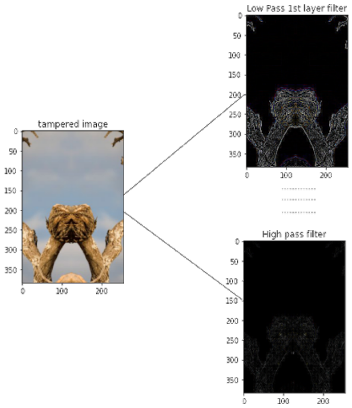


Fig. 6: SRM Filters

then merged together by either taking the mean of maximum value of each element of the feature vectors represented. Finally, each image is represented by a single vector of 400D(github).

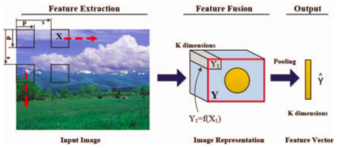


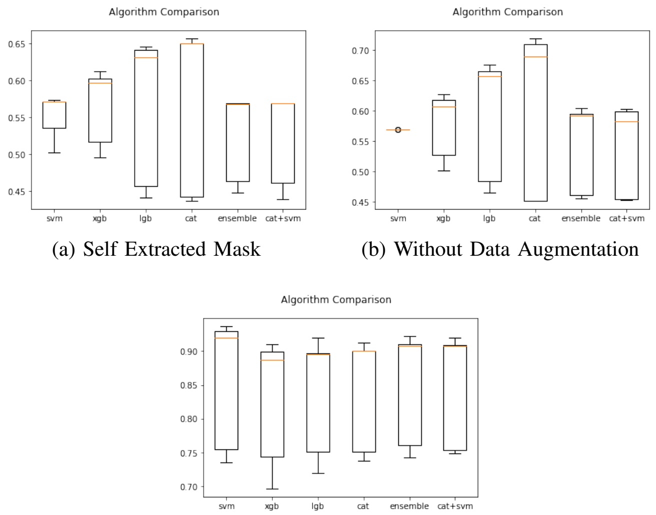
Fig. 7: Feature Fusion

V. Results

We have used pytorch for implementing the architecture of CNN. The performance extracted 400 features was evaluated on different classifiers using 10 fold cross validation and the evaluation metric used is accuracy. The analysis has been done on the following three approaches:

- Tampered regions detected using self extracted masks.
- Tampered regions detected using ground truth masks without data augmentation(4 patches for CNN Training).
- Tampered regions detected using ground truth masks with the data augmentation(8 patches for CNN training)

CNN training is done on patches generated by above 3 approaches, on parameters- epochs=250, learning-rate=0.0001, batch-size=200. Min Loss Achieved =0.3 shown in Fig 7. on Data Augmentation approach. After feature extraction, classification is done on 6 classifiers - SVM, CatBOOST, XGBoost, LightGBM, StackingClassifier(layer 1- SVM, CatBoost, XGBoost, LGBM, layer 2- Logistic Regression) , StackingClassifier(layer 1-SVM, CatBoost, layer 2- Logistic Regression). We have also checked accuracy on CASIA1 dataset after training CNN on CASIA2 patches , accuracy reported was 57.8%,57%,62% respectively on the three CNN models. BoxPlot Analysis is shown in Fig.8. Best Validation Accuracy is reported on SVM - 85.7% on tuned Hyperparameter C=1000 , Gamma=0.0001 Fig 10. Final Accuracy achieved 93% on SVM Classifier.



(a) Self Extracted Mask

(b) Without Data Augmentation

(c) With Data Augmentation

Fig. 8: BoxPlot Analysis of Classifiers

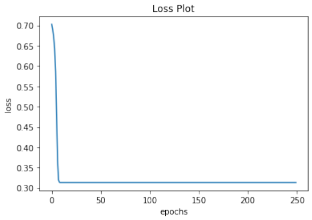
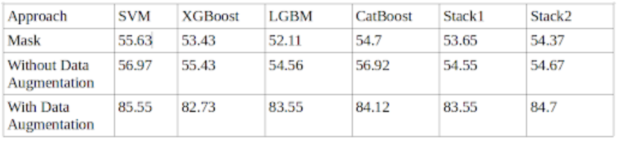
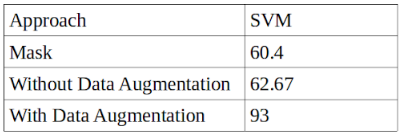


Fig. 9: CNN Training on Augmented Data



Approach	SVM	XGBoost	LGBM	CatBoost	Stack1	Stack2
Mask	55.63	53.43	52.11	54.7	53.65	54.37
Without Data Augmentation	56.97	55.43	54.56	56.92	54.55	54.67
With Data Augmentation	85.55	82.73	83.55	84.12	83.55	84.7

Fig. 10: Validation Accuracies

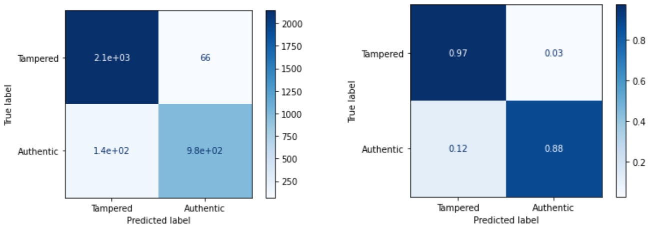


Approach	SVM
Mask	60.4
Without Data Augmentation	62.67
With Data Augmentation	93

Fig. 11: Accuracies on SVM classifier

VI. Infrences

The confusion matrix is result of the model which has achieved the highest accuracy. It can be seen that most of the tampered images are recognised appropriately.

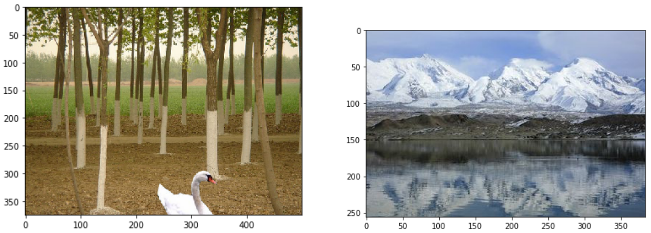


(a)

(b)

Fig. 12: Confusion Matrix of Best Approach(3)

Following are the few misclassified images by the best model obtained. The reason we can understand behind it that the mountains image of mountains has a very little tampering therefore the tampered regions were not properly detected. In the first image of forest, proper smoothening has been done which failed our model to detect the tampered region.



(a)

(b)

Fig. 13: Misclassified Images

Future work could be to detect more type to forgery. More robust CNN architecture can be developed so that it can detect the blurred tampering and illumination images.

VII. Individual Contribution

After project inception, we read up on research papers and medium articles to obtain information on how we can do CNN based Forgery detection route for the project along with our future plans. The writing of project proposals, presentations and this final report has been a joint effort. We tackled the this problem with two different approaches and each team member took one approach. The model training with ground truth images has been done by Bhakti and the model training along with self extracted masks was done by Anmol. While we coded on separate solutions to the same end result, we always discussed the problems we faced and the approaches we should make in our respective problems. Each team member is aware of the intricacies of their own problem and simultaneously has a good grasp on the whole project as a whole.

References

[1] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1053–1061, 2018.

[2] Yuan Rao and Jiangqun Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. In 2016 IEEE International Workshop on In- formation Forensics and Security (WIFS), pages 1–6. IEEE, 2016.

[3] Belhassen Bayar and Matthew C Stamm. A deep learning approach to universal image manipulation de- tection using a new convolutional layer. In Proceed- ings of the 4th ACM Workshop on Information Hiding and Multimedia Security, pages 5–10. ACM, 2016.

[4] Weiqi Luo, Jiwu Huang, and Guoping Qiu. Robust detection of region-duplication forgery in digital im- age. In Proceedings of the 18th International Confer- ence on Pattern Recognition-Volume 04, pages 746– 749. IEEE Computer Society, 2006.

[5] P. Zhou, X. Han, V. I. Morariu and L. S. Davis, "Learning Rich Features for Image Manipulation Detection," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 1053-1061, doi: 10.1109/CVPR.2018.00116.

[6] https://medium.com/@vvsnikhil/image-forgery-detection-d27d7a3a61d