INDIAN INSTITUTE OF ENGINEERING SCIENCE AND TECHNOLOGY ,SHIBPUR

DEEP NEURAL NETWORKS FOR DIGITAL FORENSIC APPLICATIONS

1856 १८५६

उत्तिष्ठत जाग्रत प्राप्य वरान् निबोधत

INDIAN INSTITUTE OF ENGINEERING SCIENCE AND TECHNOLOGY, SHIBPUR

भारतीय अभियांत्रिकी विज्ञान एवं प्रौद्योगिकी संस्थान, शिवपुर

By,

Siddhartha Ghosal (510818006)
Sagnik Dutta (510818049)
Upamanyu Mukharji (510818025)

Under the guidance of
Dr. Ruchira Naskar

A report submitted in fulfillment of the requirements of the mini project of Bachelor of Technology in Information Technology (5th Semester)

Department of Information Technology
Indian Institute of Engineering Science and Technology ,Shibpur
P.O Botanic Garden , Howrah 711103 , WB , India

January, 2021

CONTENTS

ACKNOWLEDGEMENT

We would hereby like to avail this opportunity to extend our gratitude to our project guide Dr. Ruchira Naskar, Assistant Professor, Department of Information Technology, Indian Institute of Engineering Science and Technology, Shibpur, for her supervision and valuable advice throughout the entire period without which it would not have been possible to complete the portion of the project scheduled for the semester in time.

I would also like to thank Dr. Sukanta Das, Associate Professor and Head of the Department, Department of Information Technology, Indian Institute of Engineering Science and Technology, Shibpur, as well as other faculty members and staff members for their support and help us when required.

Finally, we cordially thank our fellow classmates of Information Technology, of the Indian Institute of Engineering Science and Technology, our parents and all others who have been attached to this phase of the development of the project directly or otherwise.

INTRODUCTION

With the advent of social networking services such as Facebook , Instagram , there has been a huge increase in the volume of image data generated in the last decade.

As an expected outcome,huge quantities of fake images are being circulated in these websites everyday to incite violence , affect public opinion and bypass security checks.

These ***tampered*** images and/or videos are mainly created using image and video processing softwares such as GNU Gimp , Adobe Photoshop, etc. They pose a major concern for internet companies as well as for individuals for reasons mentioned above.

Before action can be taken on the basis of a questionable image , we must verify its authenticity.So as a result we need to develop a technique to get the knowledge as to whether an image is manipulated or not.

So, here comes the concept of convolutional neural networks. These are neural networks commonly used for analyzing visual imagery. Convnets have few important properties which makes them super efficient regarding image processing. They operate on local patterns of images. Images can be broken down into edges, borders, textures and so on. Patterns learnt by covnets are translation invariant as well as they can learn spatial hierarchies of data.

One of the key characteristics of digital image is it is easier to manipulate in comparison to its conventional counterpart. The process of digital image tampering is made easier due to the large number of available softwares. One of the most tricky techniques is the art of image splicing. Splicing is the process of making a composite picture by cutting a portion of an image and embedding it into a different image. The copied portion can go under different processing such as rotation, scaling, etc. All these are done to manipulate the image as much as possible and are extremely difficult to detect. Sufficient research in image processing has evidence showing accuracy of work of image splicing with CNN is much higher compared to any other methods. Thus finding fake images especially related to splicing with techniques like CNN can be easily achieved.

But merely knowing that the image is fake or not isn't sufficient. There may be cases where we need to know which part of the image is **spliced.** This discussion brings us to the heart of the problem that is :

"Localization of splicing in digital images".

PROBLEM MODEL

Our problem is to localize the region of forgery in the image. ***Localization*** in deep learning refers to the drawing of a boundary to represent a region of interest in a picture. It can be considered as a step above image classification. We have already taken that the images are **tampered,** we have chosen to work on identifying the location of the tampering. Hence our problem takes classification as 'tampered' by default and we work on finding out exactly where the forgery of the image has taken place.

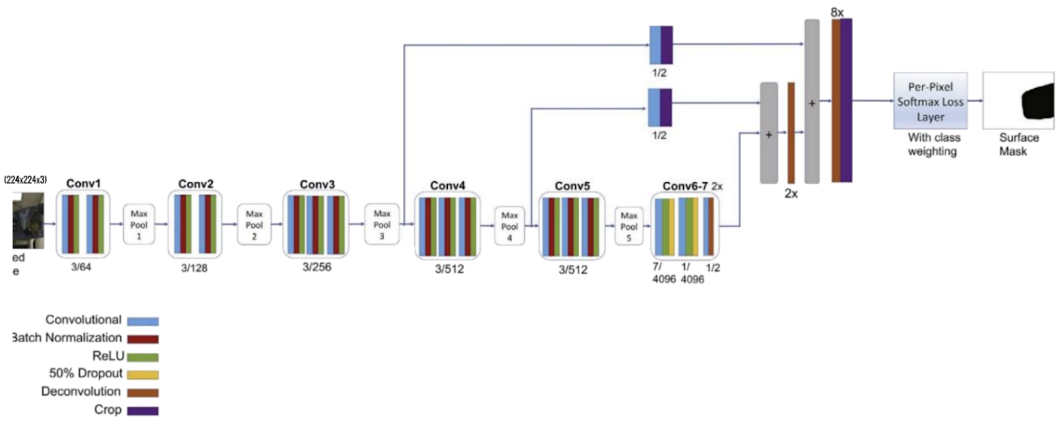
Let's take an example from CASIA 1.0 dataset.

If we need to determine which part of this image is tampered; we have to somehow generate the ground truth mask of this image.As we can definitely see the spliced region is localized by the white part in the image below. This forms what is known as a **surface mask ,** in which the tampered part of an image is given a different colour to mask or differentiate it from the rest of the genuine image.

So,if we develop some kind of deep learning based *technique* to generate this ground truth mask from a given tampered image then we can easily solve our problem. An appropriate CNN (convolutional neural network) model will help us achieve this result.

SOLUTION APPROACH

To solve this problem we have used a deep learning architecture called Fully convolutional neural network (FCN) proposed in [1]. We have used the variation of the following architecture with some minor modification, that is ,we have passed the output of the last layer in a convolutional layer with 1 filter and 1x1 kernel size in order to reduce the number of channels in the output image to 1.



We haven't used the edge-enhanced technique mentioned in [1] , instead we proceeded with the surface mask only.

The speciality of this architecture is that it has no fully connected layer . Apart from that it can be seen that the modified output of max pooling layer 3 & 4 have been fed into some other layer in future in the model,these are called *skip connections*. These connections play a very important role in image segmentation tasks.

We have taken the **VGG16** pretrained on the ***imagenet*** dataset [5] and then we extract some layers out of it and added additional layers like ***Batch Normalization*** layers and ***ReLu*** layers and skip connections to create the following model specified in **Model Summary**.

For the architecture to be modeled in code we have used Python 3.8 as the coding language.The libraries we have used to code it in Python are :

1) Keras
2) Numpy
3) OpenCv
4) Scikit learn

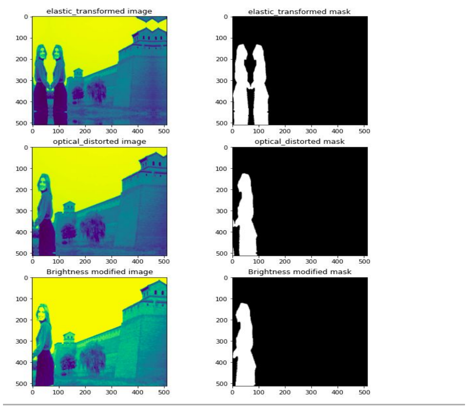
Also ,the most important thing to note is, all of our work has been done on the **Tensorflow 2.0** framework developed by Google.

EXPERIMENTAL DATA

To train the model we have used the CASIA 1.0 dataset whose corresponding surface masks were available at [6].Then We have augmented the image and its mask using three methods:

1) Random Brightness & Contrast
2) Optical Distortion
3) Elastic Transformation

All of which are provided by the ***albumentation*** library in Python 3.8 .



The reason behind this is that spliced portions of images have differences in illumination, brightness, contrast and stretch to make it look original.

EXPERIMENTAL PARAMETERS

Then we initially set to train it for 10 epochs with callbacks such as ***Early stopping*** with **patience** = 3 and **monitor = 'loss'** and ***Batch size = 20***. But after 4 epochs the ***Early stopping*** stopped the training process. We have set the learning rate to **1e-3 .** And as an optimizer, we have used ***Adam.***

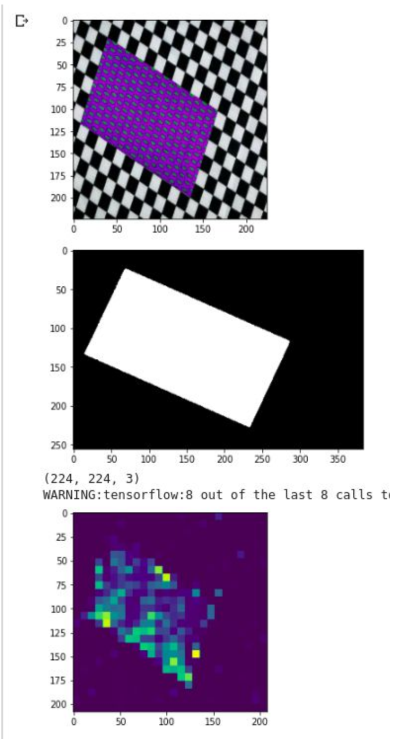
```
[9] es_cb = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=3)
    tb_cb = tf.keras.callbacks.TensorBoard('./logs', update_freq=1)
    history = MFCN.fit(X,Y,batch_size=20,epochs=10,verbose=1,callbacks=[es_cb,tb_cb])
    MFCN.save("/content/drive/MyDrive/dataset/my_model_1.h5")

    Epoch 1/10
    22/22 [==============================] - 1029s 47s/step - loss: 0.0554 - accuracy: 0.8806
    Epoch 2/10
    22/22 [==============================] - 1027s 47s/step - loss: 0.0559 - accuracy: 0.8797
    Epoch 3/10
    22/22 [==============================] - 1032s 47s/step - loss: 0.0557 - accuracy: 0.8799
    Epoch 4/10
    22/22 [==============================] - 1034s 47s/step - loss: 0.0551 - accuracy: 0.8811
```

EXPERIMENTAL RESULTS

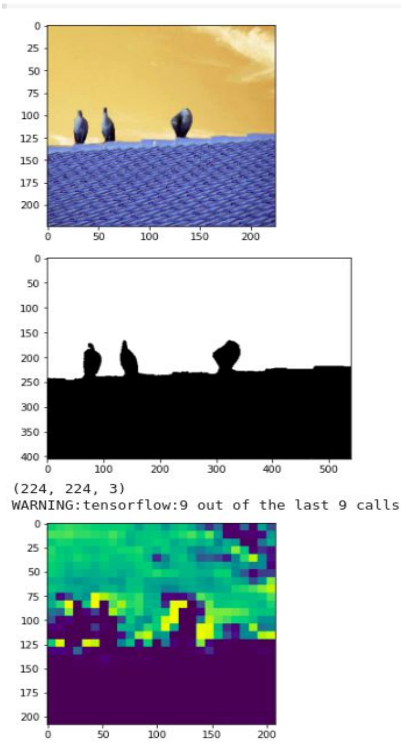
After the completion of training, We have tested the model on images from CASIA 2.0 dataset whose results are shown below with the spliced images ,corresponding ground truth masks and predicted ground truth masks.

As it can be seen in the third image of each result set that the spliced part has been localized to a certain region in the image corresponding to the spliced part in the originally spliced image that is the first image of each result set.Also our predicted mask is a close approximation of the ground truth mask

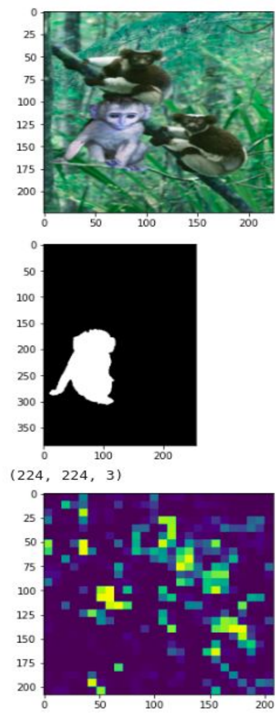
Result set 1 :



Result Set 2 :



Result Set 3:



CONCLUSION

It can be seen from the *Experimental Results* that we have successfully solved the problem of localizing the spliced part, although to a smaller extent, correspondingly all of our code is available at image splicing localization repository .

However there are certain observations , we made during the experimentation :

- **Improvement of accuracy :**

  The model could have predicted better results :

1. If we could have used a bigger dataset for training but the problem, we faced, was inadequate hardware resources to train such a big dataset.

2. Our initial training of the model was with only 886 images. so it was not that much of a quantity compared to what is required to improve the accuracy of the model significantly.

3. There is a scope of improving accuracy by introducing the *edge enhanced* technique mentioned in [1].

- **Future Scope of work :**

  After improving the accuracy, we can deploy this model so that people can easily identify the originality of an image and also the part where it has been forged.This will allow necessary mechanisms to be put in place where it will stop spreading of fake news based on forged images.

REFERENCES

[1] Ronald Salloum, Yuzhuo Ren, C.-C. Jay Kuo, Image Splicing Localization using a Multi-task Fully Convolutional Network (MFCN), Journal of Visual Communication and Image Representation, Volume 51,2018

[2] online python documentation

[3] keras online documentation

[4] tensorflow python documentation

[5] VGG 16 Architecture

[6] CASIA Dataset ground truth repository

[7] python albumentation library