**VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI-590018**

****

**B.L.D.E. A’S V.P. Dr. P.G. HALAKATTI COLLEGE OF ENGINEERING AND TECHNOLOGY, VIJAYAPUR – 586 103**

**bldeacet_header_logo_caps.png**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**A PROJECT REPORT ON**

**"**IMAGE SPLICING FORGERY DETECTION USING DEEP LEARNING**"**

**UNDER THE GUIDANCE**

**OF**

**Prof. PRABHU R BEVINAMARAD**

**SUBMITTEDBY:**

PUSHPA JAGGAL (2BL17CS063)

PADMA B SASNUR (2BL17CS045)

SHWETA BIRADAR (2BL17CS083)

RESHMA KUNABI (2BL17CS068)

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI-590018**

****

**B.L.D.E. A’S V.P. DR. P.G. HALAKATTI COLLEGE OF ENGINEERING AND TECHNOLOGY, VIJAYAPUR – 586 103**

**bldeacet_header_logo_caps.png**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that the Project[17CSP85] on work entitled “IMAGE SPLICING FORGERY DETECTION USING DEEP LEARNING” is a bonafide work carried out by PUSHPA JAGGAL (2BL17CS063), PADMA B SASNUR(2BL17CS045), SHWETA BIRADAR(2BL17CS083), and RESHMA KUNABI(2BL17CS068) submitted in partial fulfilment for the VIII Semester of Bachelor of Engineering degree of the Visvesvaraya Technological University, Belgaum during the year 2020-2021. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The seminar report has been approved as it satisfies the academic requirements in respect of project work prescribed for project prescribed in Computer Science and Engineering of VIII semester.

**GUIDE HOD PRINCIPAL PROF.PRABHU R BEVINAMARAD DR. PUSHPA PATIL DR. ATUL AYARE EXAMINERS SIGNATURE**

**1.**

**2.**

**ACKNOLEDGEMENT**

I am glad to have this great opportunity to thank all the people who helped, guided and co-operated to complete project named **"**Image splicing forgery detection using deep learning**"** with a great success.

I would like to express deep sense of gratitude to our beloved principal **DR. ATUL AYARE for** providing all facilities in the college.

I would like to thank our Head of Department **DR. PUSHPA B. PATIL** for providing facilities and fostering congenial academic environment in the college.

I feel deeply indebted to our esteemed guide **Prof. Prabhu R Bevinamarad** and for motivating and guiding us throughout this project.

I would take this opportunity to thank all the faculty members and supporting staff for helping us in this endeavour.

**ABSTRACT**

Image forensics is an active research area due to the large number of shared images online. These images can be easily manipulated with advanced image editing tools and the changes cannot be captured easily by bare human eyes. In this paper, a novel deep learning Convolution Neural Network model is proposed to extract appropriate features of an image in order to train the model efficiently and detect image splicing forgeries. The proposed technique is tested against Columbia Uncompressed Image Splicing Detection publicly available data set to test the validity of the proposed technique. Based on the result obtains the proposed technique is able to detect/classify the given testing image undergo any splicing forgery or not.

**CONTENTS**

**Page**

1. Introduction 01

2. Literature Survey 03

3. Existing System 04

4. Work Flow Diagram 05

5. Methodology 06

6. Requirements 12

7. Implementation 13

8. Result findings and analysis 17

9. Conclusion 21

10. References 22

**1. INTRODUCTION**

Digital images have become an integral part of our daily life because they can convey rich information. Due to the wide use of image acquisition tools, which are now very popular in mobile phones, the volume of images shared online is growing exponentially. However, there are many professional software packages, such as Photo shop and Coral-draw that can be used to manipulate images and alter their contents without leaving visible artifacts for human eyes. Two common approaches for image manipulation/forgery are: copy-move and splicing. In copy-move forgery, some parts of an image are copied to other regions in the same image. By contrast, image splicing combines two or more different images into one image.

Figure 1 shows an illustrative example of image splicing where the American General Francis P. Blair was added to the Mathew Brandy’s photo although he didn’t actually attend this meeting. It is obvious that image forgery can have drastic effect on the authenticity of images. Due to the advancement in image forgery methods, a tampered region of an image is hard to detect with bare human eyes. It has become crucial to develop automated methods to detect more sophisticated image forgeries in the large number of available images. In the literature, many methods have been proposed for detecting image forgery. These methods can be either active or passive. Active techniques use verification of signatures pr-embedded in the image, e.g., watermarking or stenography.



Figure 1. Francis P. Blair (right) was added to Mathew Brady’s famous photo [2]

The remainder of this paper is organized as follows. Section I briefly reviews related work. Section II describes the existing system and its issues Section III explains the work flow diagram Section IV describe the proposed methodology including feature extraction and classification. Section V explains the hardware and software requirements. Section VI describes the implementation details. Section VII result finding and analysis .Section VIII concludes the project.

**2. RELATED WORK**

Shi and Chen [3] proposed a method using statistical features from a 2D array including wavelet and Markov features. Images were divided into multiple blocks then a multiple block discrete cosine transform (MBDCT) was applied. Zhang et al. [4] used DCT coefficients and LBP histogram as a feature vector to detect image splicing. To reduce the complexity and features dimensionality, Principle Component Analysis (PCA) was used. The same authors in [5] proposed another model which shows the statistical difference between authentic and spliced images. This approach used MBDCT and statistical image quality metrics which includes mean square errors, normalized cross correlation and moment-based features. Wang et al. [6] used edge information of image chrominance component using a finite-state Markov chain. They extracted a low-dimensional vector which was fed into a Support Vector Machine (SVM) to detect image authenticity. This work has been extended in [7]. Markov features have been extracted by using transition probability matrices in DCT and Discrete Wavelet Transform (DWT) domains. The recursive feature elimination (RFE) SVM was used. Zhao et al. [8] explored the chroma channels and four gray level run-length run-number (RLRN) vectors in various directions were extracted as features.

**3. EXISTING SYSTEM**

Many methods have been proposed for detecting image forgery. These methods can be either active or passive. Some of the issues of existing systems are.

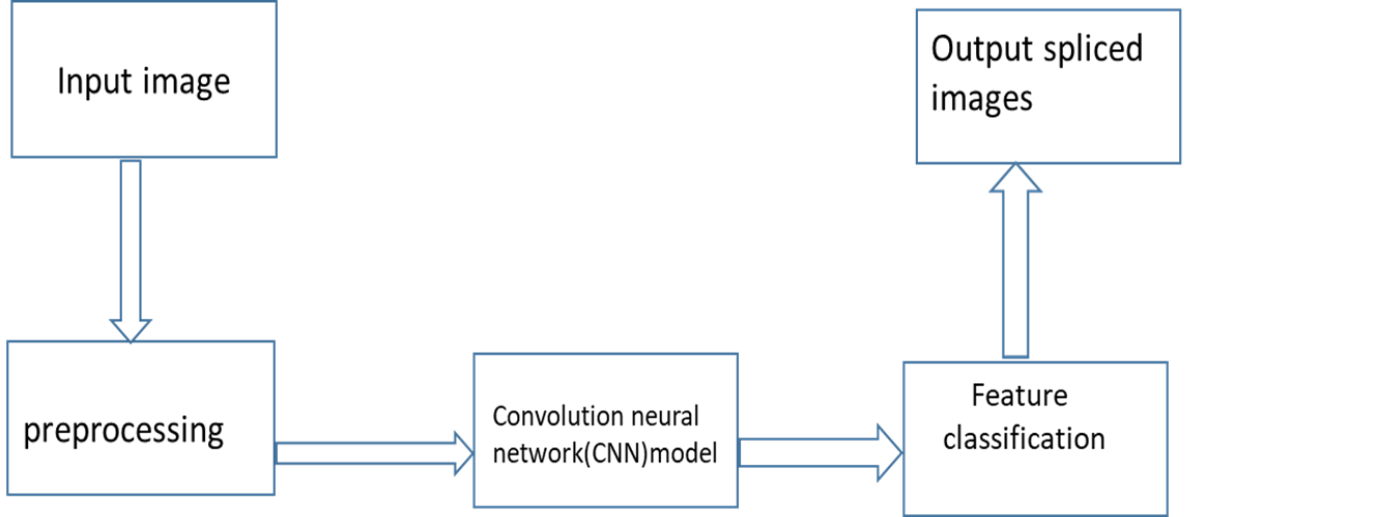
**3.1. ISSUES**

The existing system which includes the following issues.

* Active techniques use verification of signatures pr-embedded in the image, e.g., watermarking or stenography.
* Some of the camera models have no such facility to insert signatures which is one of the disadvantages of active techniques.
* On the other hand, passive techniques analyze the digital image characteristics to track the forged regions in the image [1].
* Most of the work in the literature has focused on copy-move forgery.
* There are some methods for detecting image splicing but in the general the accuracy is relatively lower.

**4. Work flow diagram**

* A workflow diagram is a step-by-step, linear representation of process from start to finish
* It shows how an individual task, actions, or resources flow between the modules.

****

**Figure 2. The workflow of image splicing detection**

The workflow of image splicing detection is shown in figure 2. The first step in this workflow is preprocessing, which is an optional step. Various image preprocessing operations used in past include RGB to YCbCr conversion [8], [9], RGB to HSV conversion [10], and RGB to Grayscale conversion [11]. The main purpose of the preprocessing step is to focus on the features related to a specific channel. The feature extraction step in figure 2 is the most important step where useful features are extracted from the input image. In the early days of image forgery detection, the methods mainly relied on the traditional feature extraction techniques used in computer vision. Sometimes these feature extractors are known as hand-crafted feature extraction techniques as these features are primarily designed to focus on some specific characteristics of the image and created manually by some experts. Some of the popular hand-crafted feature extractors that were used in image splicing detection are, discrete wavelet transform (DWT) [12], contourlet transform [14], Hilbert-Huang transform [15], local binary pattern (LBP) [13], discrete cosine transform (DCT) [13]. Mostly hand-crafted feature-based image splicing detectors use a support vector machine (SVM) as a classifier [12], [13]. As opposed to hand-crafted feature-based image splicing detection methods the recent developments ([16], [17], [18], [19], [20]) have focused on deep learning-based image splicing detection. In general, the deep learning-based methods can learn more generalized features from the input images. Hence, deep learning-based image splicing detection methods have become much popular in the last few years. In this way, all the existing image splicing detection methods can be classified as hand-crafted feature-based and deep learning-based method.

**5. The Proposed Methodology**

We proposed a deep learning Convolution Neural Network model is to extract appropriate image features to train the CNN model efficiently and detect image splicing forgeries. The proposed technique contains five modules to implemented using python programming language. Finally the proposed model is tested against Columbia Uncompressed Image Splicing Detection publicly available data set to test the validity of the proposed technique. Based on the result obtains the proposed technique is able to detect/classify the given testing image undergo any splicing forgery or not. The proposed methodology contains following modules which is explained in the below subsections.

**Project modules**

1. Data collection

2. Data-preprocessing and augmentation

3. Data cross-validation

4. Building deep learning module (Convolution Neural Network)

5. Training and testing data

6. Result findings and analysis

**1. Data collection: Columbia Uncompressed Image Splicing Detection**

The required standard dataset has been downloaded from https://www.ee.columbia.edu/ln/dvmm/downloads/authsplcuncmp/#:~:text=The%20image%20sizes%20range%20from,%2C%20books%20...et.

## Introduction to dataset.

Copying-and-pasting, or image splicing, is the most common tampering seen today. Although often followed by various post processing techniques, we provide a benchmark set with only the splicing operation so that people can study its effect in a focused way. Our images are in high resolution and uncompressed, removing further the compression concern.

* **Image Content**

There are 2 directories in this dataset: 4cam\_auth & 4cam\_splc. 4cam\_auth contains authentic images, and 4cam\_splc contains spliced images. By the term 'authentic', we mean an image that is taken using just one camera. In 4cam\_auth, there are 183 images, and in 4cam\_splc, there are 180. The image sizes range from 757x568 to 1152x768 and are uncompressed, in either TIFF or BMP formats. The spliced images are created using the authentic images. Only 27 images, or 15%, are taken outdoors on a cloudy day (which makes the outdoor illumination similar to indoor conditions). Several examples are shown below.

****

**Figure 3. Example images in dataset**

**2. Data-preprocessing and augmentation**

In order to help the model to extract significant features . We have preprocessed data such as reduced the size of an image into 300x300 pixels and we have augmented the data to artificially expand the size of a training dataset by creating modified versions of images in the dataset to improve the ability to detect the images correctly.

* Rescaled the image pixel value using (rescale=1. /255).
* Brightness change
* Horizontal flip etc.

**3. Preparation for data cross-validation**

Cross-validation is a re-sampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into.

The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like over fitting or selection bias and to give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem).

**4. Building deep learning module.**

Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. **Deep learning** is an important element of data science, which includes statistics and predictive modeling.The following figure depicts the significant of using deep learning concept.

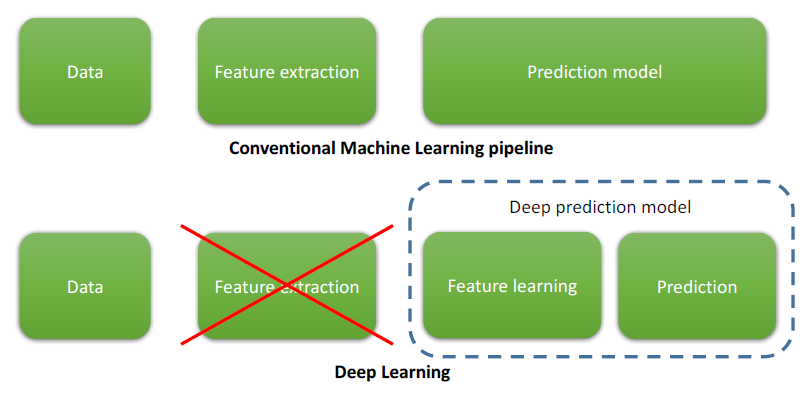


Figure 4. Deep Learning model

It automatically learns feature at multiple levels of abstraction that allows a system to learn all possible features from low-level features such as edges and pixel intensity of an image to high-level features such as objects and shapes without depending on human engineered or human extracted features.

* **Convolution Neural Networks**

A convolutional neural network, or CNN, is a deep learning neural network designed for processing structured arrays of data such as images. Convolutional neural networks are widely used in computer vision and have become the state of the art for many visual applications such as image classification, and have also found success in [natural language processing](https://deepai.org/machine-learning-glossary-and-terms/natural-language-processing) for text classification.

Convolutional neural networks contain many convolutional layers stacked on top of each other, each one capable of recognizing more sophisticated shapes. With three or four convolutional layers it is possible to recognize handwritten digits and with 25 layers it is possible to distinguish human faces.

## The architecture of a convolutional neural network is a multi-layered feed-forward neural network, made by stacking many hidden layers on top of each other in sequence. It is this sequential design that allows convolutional neural networks to learn hierarchical features. The hidden layers are typically convolutional layers followed by activation layers, some of them followed by pooling layers. The following figure depicts the CNN architecture.

## https://www.upgrad.com/blog/wp-content/uploads/2020/12/1-4.png

## Figure 5. Architecture of a Convolution Neural Network

1. A convolution tool that separates and identifies the various features of the image for analysis in a process called as Feature Extraction

2. A fully connected layer that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages.

### **Convolution Layers**

### There are three types of layers that make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, a CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function which are defined below.

### **1. Convolutional Layer**

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM).

The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

### **2. Pooling Layer**

In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. Depending upon method used, there are several types of Pooling operations.

In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer.

### **3. Fully Connected Layer**

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture. In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place.

### **4. Dropout**

Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model’s performance when used on a new data. To overcome this problem, a dropout layer is utilized wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

**5. Activation Functions**

Finally, one of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network. It adds non-linearity to the network. There are several commonly used activation functions such as the ReLU, Softmax, tanH and the Sigmoid functions. Each of these functions have a specific usage. For a binary classification CNN model, sigmoid and softmax functions are preferred an for a multi-class classification, generally softmax us used.

### **6. Requirements**

* A Software requirements specification (SRS) document describes the intended purpose, requirements, and nature of software/application/project to be developed.
* To prepare an SRS document, we would need to have a functional knowledge of our project or application, knowledge of software/hardware/technology to be used.

### The following are the hardware and software requirements used for implementations.

**Hardware Requirements:**

* + - Processor: Intel Corei3 processor
    - RAM: 4GB
    - Hard disk :1TB HDD

**Software Requirements:**

* Operating system: window 64bit
* Tools used: Googlecolab
* Programming language: python

**7. Implementation details**

This section details the complete implementation of proposed. The proposed system implemented using Googlecolab and python programming language.

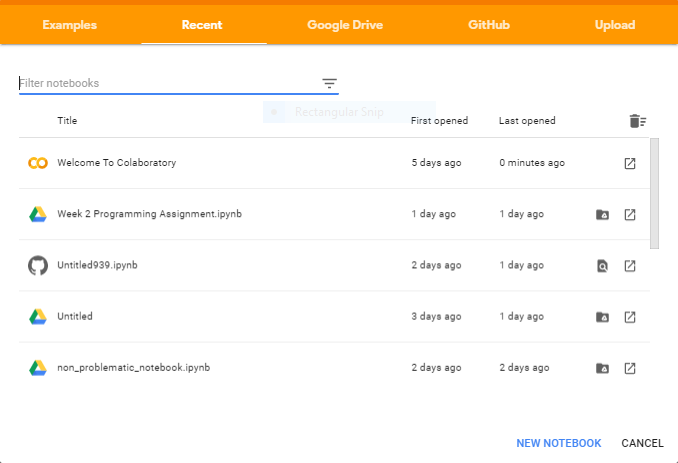
**Technology used**

**Googlecolab**

Training a deep learning model on your regular laptop uses a lot of computational power and often runs endlessly. This can dissuade beginners from personally exploring the world of deep learning.With Colab you can execute code on Google’s cloud servers by leveraging the power of Google hardware, including GPUs and TPUs, regardless of the power of your machine.The following are the different steps for creating Googlecolab note book.

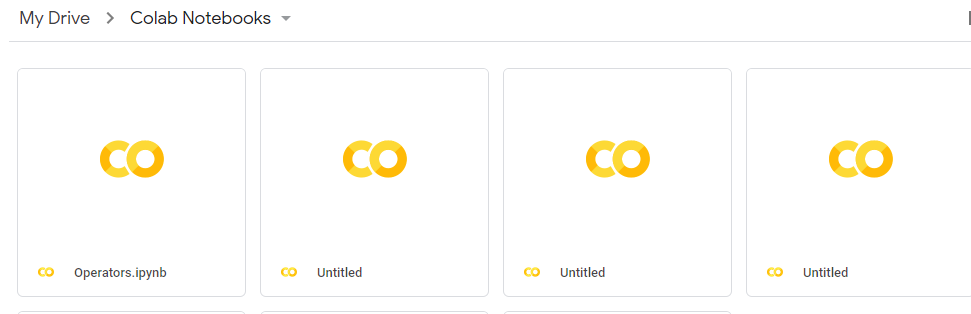
* **Creating a new notebook**

To create a new Notebook on Colab, open https://colab.research.google.com/, and it will automatically show your previous notebooks and give an option to create a new notebook.



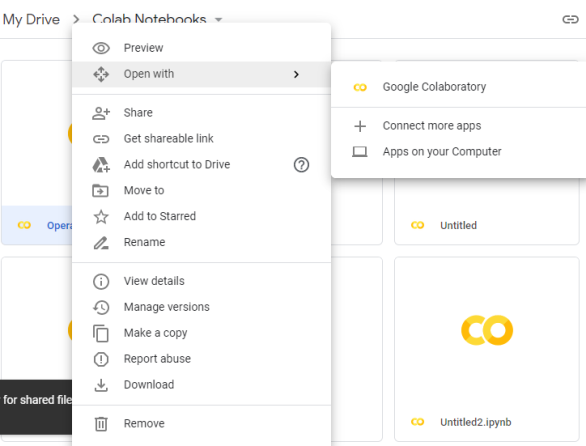
**Figure 6. Creating a new Notebook**

Here you can click on NEW NOTEBOOK to start a new notebook and start running your code in it. By default, it is a Python 3 notebook. All the notebooks you create in Google Colab, are by default stored in your Google Drive. There is a folder in your drive named “Colab Notebooks” where you can find all your notebooks created from Google Colab.



* **Opening a notebook from Drive in Colab.**

Right-click on the desired notebook and “Open With > Google Colaboratory”.



**Figure 7.Opening a notebook from Drive in Colab.**

* **Load Data from Drive**

You can easily load data from Google Drive by mounting it to the notebook. To do this, type the following code in your notebook.

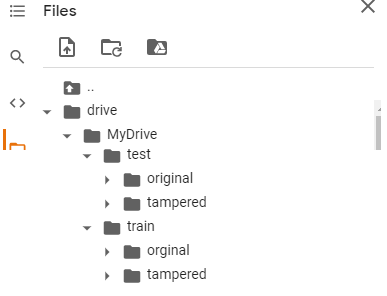
from google.colab import drive

drive.mount('/content/drive')

It will give you a link to open, then follow the below given steps

* Go to the link
* Login to your Google Account
* Copy the code
* Paste it in notebook

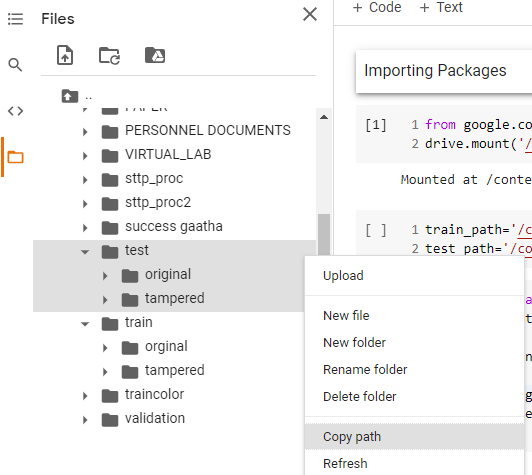
Now if you see in your “Files” section, you will find your ‘drive’ as shown below. Let’s say you have uploaded images under the my drive folder.



**Figure 8. Upload photos to drive**

**How to access the path of dataset?**

You can get the images by copying their path and pasting them to path variables.



**Figure 9.Access the path of dataset**

**Python Libraries used**

A python library is a reusable chunk of code that may want to include in our programs/projects.

Compare to languages like c++,c , a python libraries do not pertain to any specific context in python.

Here a library loosely describes a collection of core modules.

* **NumPy**

NumPy is a Python library that provides a simple yet powerful data structure: the **n-dimensional array**. The following are the various benefits of NumPy.

1. **More speed:** NumPy uses algorithms written in C that complete in nanoseconds rather than seconds.
2. **Fewer loops:** NumPy helps you to [reduce loops](https://realpython.com/numpy-array-programming/) and keep from getting tangled up in iteration indices.
3. **Clearer code:** Without loops, your code will look more like the equations you’re trying to calculate.
4. **Better quality:** There are thousands of contributors working to keep NumPy fast, friendly, and bug free.

* **Pandas**

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. Pandas make it simple to do many of the time consuming, repetitive tasks associated with working with data, including:

1. Data cleansing
2. Data fill
3. Data normalization
4. Merges and joins
5. Data visualization

* **Scikit-learn**

Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

* **Keras**

[Keras](https://keras.io/) is a high-level neural networks API developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.Keras has the following key features:

* Allows the same code to run on CPU or on GPU, seamlessly.
* User-friendly API which makes it easy to quickly prototype deep learning models.
* Built-in support for convolutional networks (for computer vision), recurrent networks (for sequence processing), and any combination of both.
* Supports arbitrary network architectures: multi-input or multi-output models, layer sharing, model sharing, etc. This means that Keras is appropriate for building essentially any deep learning model, from a memory network to a neural Turing machine.
* **Pillow (PIL library)**

The Python Imaging Library adds image processing capabilities to your Python interpreter. This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities. The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool.

**Keras deep learning neural network library**

We will use Keras deep learning neural network library which provides the capability to fit models using image data augmentation via the ImageDataGenerator class. We will also use a pre-trained InceptionV3 model for transfer learning. The system uses a Keras to load up the InceptionV3 model, and freeze the convolution blocks so that we can use it as just an image feature extractor.

**Data Pre-processing and Data Augmentation**

In order to make the most of our few training examples, we will “augment” them via a number of random transformations, so that our model would never see twice the exact same picture. This helps prevent overfitting and helps the model generalize better.In Keras this can be done via the ImageDataGenerator class. This class allows you to configure random transformations and normalization operations to be done on your image data during training instantiate generators of augmented image batches (and their labels) via .flow\_from\_directory(directory). These generators can then be used with the Keras model methods that accept data generators as inputs.

**Building Model Architecture**

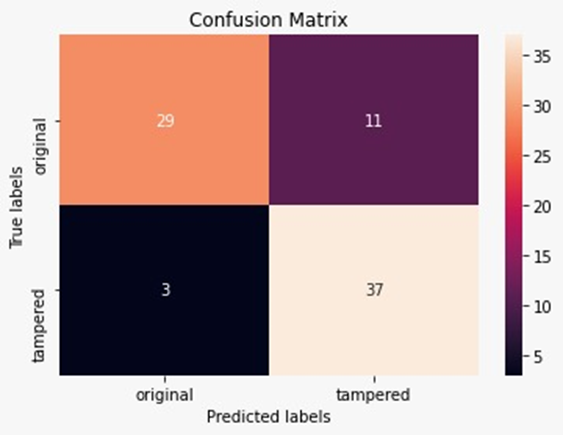
Let’s build the architecture of our deep neural network classifier now, which will take the above mentioned flattened bottleneck features as input. Training model and testing a model. We will use the architecture with checkpoints and fit the model to the training data.

**8. Result finding and analysis**

Mentioned below are the parameters on which our output depends.

**Confusion matrix**

A confusion matrix, also known as an error matrix, is a summarized table used to assess the performance of a classification model. The number of correct and incorrect predictions are summarized with count values and broken down by each class.



**Figure 10. Confusion matrix**

**Performance matrics**

Performance matrics are used to measure the behavior ,activities ,and performance metrics. Various methods for image segmentation are considered and there different performance matrices are compared.

**Precision**

Precision is also known as positive predictive value and is the proportion of relevant instances among the retrieved instances. In other words, it answers the question “What proportion of positive identifications was actually correct”

Precision=TP/TP+FP

=29/29+1

= 29/40

=72

**Recall**

Recall, also known as the sensitivity, hit rate, or the true positive rate (TPR), is the proportion of the total amount of relevant instances that were actually retrieved. It answers the question “What proportion of actual positives was identified correctly”

Recall=TP/TP+FN

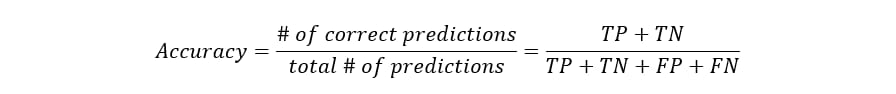
=29/29+3

=29/32

=90

**Accuracy**

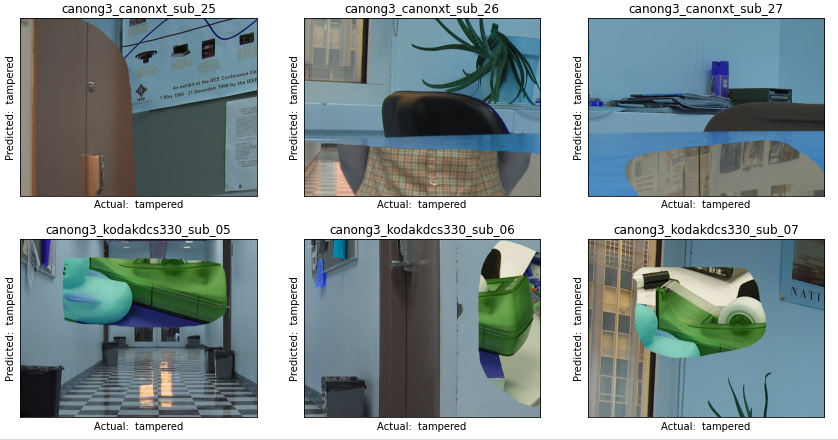
This is simply equal to the proportion of predictions that the model classified correctly.

   =29+37/29+11+3+37

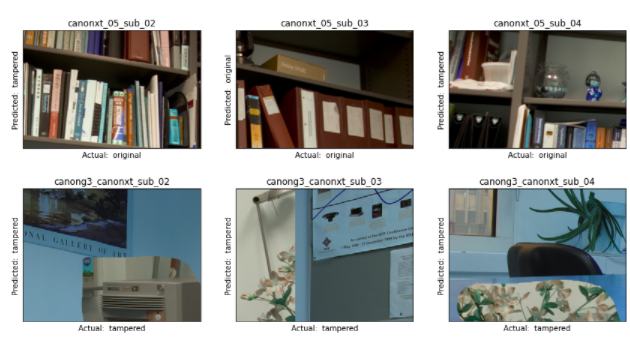
=66/80

=82

**Sample output**



**Figure 11. Detection result sample output 1**

 **Figure 12. Detection result sample output 2**

**CONCLUSION**

Image forensics is an active research area due to the large number of shared images online. These images can be easily manipulated with advanced image editing tools and the changes cannot be captured easily by bare human eyes. In this paper, a novel deep learning Convolution Neural Network model is proposed to extract appropriate features of an image in order to train the model efficiently and detect image splicing forgeries. The proposed technique is tested against Columbia Uncompressed Image Splicing Detection publicly available dataset to test the validity of the proposed technique. Based on the result obtains the proposed technique is able to detect/classify the given testing image undergo any splicing forgery or not.

**REFERENCES**

[1] G. K. Birajdar and V. H. Mankar, “Digital image forgery detection using passive techniques: A survey,” Digital Investigation, vol. 10, no. 3, pp. 226–245, 2013.

[2] “Fake pictures.” <http://www.bbc.com/future/story/20121213-fakepictures-> make-real- memories/.

[3] Y. Q. Shi, C. Chen, and W. Chen, “A natural image model approach to splicing detection,” in Proceedings of the 9th Workshop on Multimedia & Security, pp. 51–62, ACM, 2007.

[4] Y. Zhang, C. Zhao, Y. Pi, and S. Li, “Revealing image splicing forgery using local binary patterns of dct coefficients,” in Communications, Signal Processing, and Systems, pp. 181–189, Springer, 2012.

[5] Z. Zhang, J. Kang, and Y. Ren, “An effective algorithm of image splicing detection,” in *IEEE* International Conference on Computer Science and Software Engineering, vol. 1, pp. 1035–1039, 2008.

[6] W. Wang, J. Dong, and T. Tan, “Image tampering detection based on stationary distribution of markov chain,” in 17th IEEE International Conference on Image Processing (ICIP), pp. 2101–2104, 2010.

[7] Z. He, W. Lu, W. Sun, and J. Huang, “Digital image splicing detection based on markov features in dct and dwt domain,”Pattern Recognition, vol. 45, no. 12, pp. 4292–4299, 2012.

[8] X. Zhao, J. Li, S. Li, and S. Wang, “Detecting digital image splicing in chroma spaces,” in International Workshop on Digital Watermarking, pp. 12–22, Springer, 2010.

