International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Image Forgery Detection Using CNN

Dr.V.Jayapradha [1], M.Reddy Kumar [2], K.Vamsi [3], N.V.Subba Reddy [4]

[1]AssistantProfessor, [2,3,4] Student (B.E),
Department of Electronics and Communication Engineering.
Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, SCSVMV[Deemed to be
University],Kanchipuram,Tamil Nadu,India.
Jpece@kanchiuniv.ac.in, 11189C132@kanchiuniv.ac.in,11189C127@kanchiuniv.ac.in, 11189C163@kanchiuniv.ac.in

ABSTRACT

Uploading their photographs on social media has become a method of socialism and connectivity in today's world. The availability of personal pictures on social media has also made them easily accessible to others, who can modify those images. Attackers can now easily manufacture forged images thanks to recent advances in media generating tools. As a result, the idea image forensic was brought into play to verify the authenticity of these photos. Traditional image forgery detection systems require a long time to uncover forgeries. A deep neural network technique is used in the new emerging technologies for detecting picture counterfeiting. This work employs a mixed deep learning and machine learning approach to detect passive image counterfeiting. This suggested technique uses a convolutional neural network to classify photos into forged and non-forged categories, making it impossible for humans to notice the animal was inserted.

Keywords- CNN,ComputerGenerated Images,ReLu,AGC,RNN

Introduction

Images have begun to play an increasingly important role in today's society. It's amazing how we can capture a moment in time and completely freeze it in the form of an image. Unlike in the past, when photographs had to be painted or the quality of the image we could collect was inadequate, we are now seeing the development of camera sensors that can capture photos in incredible resolutions that leave us breathless with the intricacies. While most individuals take most of their photos using their phones, many people have succeeded in making photography their own expertise and pursuing it as a career. This advancement in the modern world has resulted in the development of a variety of picture editing software programs that allow users to adjust and tweak their images in a variety of ways. While most people use this software to change the color or saturation of an image, it can also be used in a variety of different ways. Photoshop is one of the most widely used software programs in the world today, and it is used to produce stunning photos with the addition of fictional settings or objects. For example, a skilled user of this software may easily insert an animal into a photograph of you standing alone, making it impossible for others to notice the addition.

The picture forgery detection method determines whether or not the image has been modified. To determine if a particular image is forged or not, a sufficient number of features are necessary. Because existing approaches for feature extraction are based on handcrafted features or feature engineering and are not invariant to various types of transformations, geometrical, and post processing operations, features based on convolutional Neural Network (CNN) models are effective features to classify the category of the image. Furthermore, feature engineering and feature extraction are important and time-consuming tasks in today's CNNs, as deeper levels of the network contain numerous layers of neurons for processing increasingly complicated data. The most significant advantage of CNN and deep learning is that they can automatically learn appropriate features, whereas developing features manually or through feature engineering is exceedingly difficult. The main purpose of this publication is to conduct the procedure for detecting picture counterfeiting using convolutional neural networks, a subdomain of machine learning.

Computer vision applications have advanced dramatically as a result of recent developments in Deep Learning, ranging from unlocking our phones with our faces to safer self-driving cars. Artificial neural networks, notably Recurrent Neural Networks 10 (RNN) and Convolution Neural Networks, are used in the majority of recent deep learning models (CNN). The current issue is the identification of image fraud. There are various methods of image forgeries, such as retouching, slicing, copy paste, copy and move, filters, and so on, as mentioned below. Image fabrication can sometimes involve removing and adding aspects to an image. Let's look at the many forms of imager forgeries.

LITERATURE REVIEW

The authors [1],Bunk et al. proposed two forgery detection algorithms in this paper. To begin, they used DNNs and resampling features to detect the tapered region. For the location of the forgery, they used random walker segmentation. Second, when those similar features are passed through a long short-term memory, they are employed for classification (LSTM).

Tarman proposed the M- SIFT approach for detecting CMF in mirror rotated images, which is based on the enhanced scale-invariant feature transform (SIFT) method. The zones are localized with a 98 percent precision, however the time required to compute this tempering procedure is considerable.

Fengli and Qinghua developed a neural network-based approach for detecting fake power frequency generation in grids. In autonomous generation control, Fourier transforms are used to detect forgeries assaults (AGC). They looked for unusual patterns produced by forgeries using area control error (ACE)time series patterns.

Thirunavukkarasu and Kumar proposed employing a rapid retina key point descriptor for passive copy-move forged region detection (FREAK). The Harris corners are transferred to the FREAK descriptors, which are then mapped to the K-means algorithm. The technology for optical distant sensing harbour images was developed by Cheng and Meng. By learning the edge-features, a convolutional network is used to observe things such as the water, land, and ship. The network is trained with the help of edge detection networks. To begin, the edge network is trained using various extracted features from the defined segmentation network. Second, the outputs of the edge trained networks are fed into the overall model to improve it. When it comes to shape formation, this procedure is carried out by adding an edge-aware regularization to the accurate, efficient, and consistent results.

SIFT characteristics were shown to be useful in minimizing distortion in images that had been processed many times by Li and Liu in their algorithm. The traces are frequently left behind while eliminating the SIFT key points, and these are erased by injecting a high number of fake SIFT key points in the last created image to reduce distortion. The results are superior to those obtained using other methods. Using DNNs, Husain et al. suggested a technique for clearly detecting scene items. This14 method relies on pixel-by-pixel learning of pre-trained geometrical features and labeling of individual objects to accurately detect indoor scenes. Amerini et al. suggested a SIFT-based key point identification method that uses the concept of geometrical changes to localize the copymove component.

PROPOSED SYSTEM

Machine learning algorithms can be used to estimate if a particular image is faked or authentic.To make predictions, we employ Convolutional Neural Networks.

Jupiter Notebook and Spyder in Anaconda are used to implement the proposed system.

- Gathering the train and test datasets
- Data Preprocessing
- Data Modelling
- Prediction

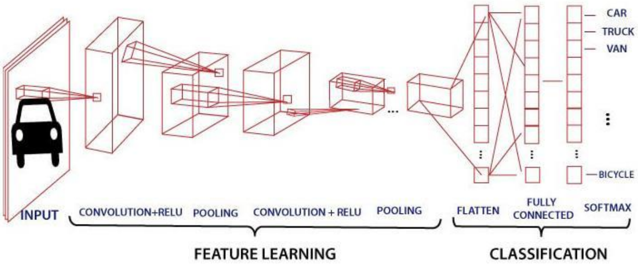
We'll need to gather the needed clean, non-tampered photos from a variety of sources, including the MICC-F220. The MICC-F220 dataset was used in our project, and it was split into two files: alex.pkl and svm.joblib. Data preprocessing is the procedure for preparing raw data for use in a machine learning model. It's the first and most important stage in building a machine learning model. The transform function in the module can be used to perform data preparation.

The model is trained, and the modules are stored in the alex.pkl and svm.joblib files. The Convolution2D layer generates a tensor of outputs by convolving a convolution kernel with the layer input. A bias vector is constructed and appended to the outputs if use bias is True. Finally, if activation is not None, the outputs are activated as well. MaxPooling2D; downsamples the input representation by taking the maximum value for each dimension along the features axis over the window provided by pool size. In each dimension, the window is adjusted by strides. When utilizing the "valid" padding option, the output has the following shape (number of rows or columns): output shape = (input shape - pool size + 1) / strides. When using the "same" padding option, the output shape is: output shape = input shape / strides. The flatten layer aids in flattening the input and has no bearing on batch size.

SYSTEM DESIGN

The design of a system provides a basic understanding of how the system is intended to function. The block diagram below depicts all of the steps involved in creating the system's design.

.



We use the convolution process to extract essential characteristics from the image in this step. We have an input matrix and a filter matrix. To create a feature map, we slide the filter matrix over the input matrix and conduct element wise multiplication.

The feature maps' dimensions are reduced by using pooling layers. As a result, the number of parameters to learn and the quantity of computation done in the network are reduced. This layer ensures that our image is both spatially and translationally invariant. Pooling can be done in a variety of ways. Max pooling, Average pooling, and Global pooling are the three options. Max pooling was applied in this case.

Pooling that selects the maximum element from the region of the feature map covered by the filter is known as max pooling. As a result, after applying the max pooling layer, the outcome would be a feature map 37 having the most prominent features from the preceding feature map.

Feature maps are converted to vector form using the Flatten layer. Flattening is the process of transforming data into a one-dimensional array that may be passed on to the next layer. To construct a single lengthy feature vector, we flatten the output of the convolutional layers. It's also linked to the final classification model, which is referred to as a fully-connected layer. The inputs to the fully connected layer come from the flatten layer. It classifies the image and suggests the optimal label for describing it.

A highly connected layer learns features from all of the previous layer's combinations, but a convolutional layer depends on consistent characteristics with a tiny repeated field. Dense layers have a unique nonlinearity property that allows them to mimic any mathematical function. They are, however, constrained in the sense that we always get the same output vector for the same input vector.

Hidden layers are prevalent in neural networks, but their function and construction vary a lot from one case to the next. Hidden layers can be distinguished by their functional features, as mentioned above. For example, a hidden layer used to identify wheels in a CNN used for object recognition cannot identify a car on its own; however, when combined with additional layers used to identify windows, a large metallic body, and headlights, the neural network can make predictions and identify possible cars within visual data.

SYSTEM REQUIREMENTS

SOFTWARE REQUIREMENTS

For developing the application, the following are the Software Requirements:18

1. Python
2. Django

Operating Systems supported

1. Windows 7
2. Windows XP
3. Windows 8
4. Windows 10

Technologies and Languages used to Develop

1. Python
2. Jupyter Notebook
3. Pycharm

Debugger and Emulator

- Any Browser (Particularly Chrome)

Hardware Requirements

For developing the application, the following are the Hardware Requirements:

- Processor: Core I5 or higher
- RAM: 8 GB
- Space on Hard Disk: minimum 500 GB

METHODOLOGY

Creating Flask:

To integrate our built model, we'll use the Flask API. Flask is a python web framework that includes features for creating web applications such as managing HTTP requests and rendering templates. Python comes with a number of web frameworks for building online apps and APIs. The most well-known is Django, a framework with a pre-defined project structure and numerous built-in utilities. For experienced programmers, this can save time and effort, but it can also be intimidating.

Flask apps are more suitable to a contained application like our prototype API because they are created on a blank canvas. Integrating Flask apps with Front-end frameworks is one of the most important components of Flask development.

Dataset preperation:

A dataset is a grouping of data. The dataset for the Convolutional Neural Network is made up of photos. The MICC-F220 publically accessible benchmark dataset is used in this study. There are 220 photos in this dataset. One hundred and ten photos are genuine, while the other one hundred and ten are fakes.

The information is divided into two categories:

- Authentic/ Unique
- Examples of tampered/fake original images

Model Building:

The CNN model is trained using the training data. The model libraries are being imported. The model is trained, and the modules are stored in the alex.pkl and svm.joblib files. The Convolution2D layer generates a tensor of outputs by convolving a convolution kernel with the layer input. A bias vector is constructed and appended to the outputs if use bias is True. Finally, if activation is not None, the outputs are activated as well. MaxPooling2D; downsamples the input representation by taking the maximum value for each dimension along the features axis over the window provided by pool size. In each dimension, the window is adjusted by strides. When utilizing the "valid" padding option, the output has the following shape (number of rows or columns): output shape = (input shape - pool size + 1) / strides. When using the "same" padding option, the output shape is: output shape = input shape / strides. The flatten layer aids in flattening the input and has no bearing on batch size.

Configure the learning process:

After we've defined our model and stacked the layers, we'll need to configure it. During the compilation step, we perform this configuration process. We must first assemble the model and define the loss function, optimizers, and prediction metrics before we can train it. We use the.compile() method to compile the model. The compile() method in the Keras model is used to compile the model.

Training the model:

NumPy arrays with fit are used to train models (). This fit function's main aim is to evaluate your model during training. This can also be used to graph the performance of a model. An epoch is a word used in machine learning that refers to the number of runs the machine learning algorithm has made across the full training dataset39. Batches are commonly used to group data sets (especially when the amount of data is very large). Before declaring one epoch ended and starting the next, samples per epoch can be the total number of steps (batches of samples) to yield from the generator. It's usually calculated by dividing the number of unique samples in your collection by the batch size. Determining how many epochs a model should run to train is dependant on a number of parameters relating to both the data and the model's purpose, and while attempts to turn this process into an algorithm have been made, a thorough understanding of the data is frequently required. After your model is finished, nbval samples sets how many validation samples your model will evaluate.

CONCLUSION

CNN is a very powerful image classification and object identification technique that is widely utilized. CNN is a fairly robust technique for various image and object identification applications because to its hierarchical structure and extensive feature extraction capabilities from a picture. Because it has minimal latency and utilizes less processing resources while maintaining high accuracy, the depth-wise separable convolutional neural network model developed in this study is well suited for mobile devices. CNN is a fairly robust technique for various image and object identification applications because to its hierarchical structure and extensive feature extraction capabilities from a picture. The fundamental advantage of CNN over its predecessors is that it discovers essential traits without the need for human intervention. Among all the algorithms, CNN has the best picture classification accuracy.

REFERENCES

1. Caliskan, A., Cevik, U.:' An efficient noisey pixels detection model for CT images using extreme learning machines', The. Vjesn., 2018, 25, (3), pp. 679-686.

2. Zhao, W., Du, S., William, J.: 'Object-based convolutional neural network for high-resolution imagery classification', *IEEE J. Sel. Top. Appl. Earth Obs*

3. Jian, L., Xiaolong, L., Bin, Y., *et al.*: 'Segmentation-based image copy-move forgery detection scheme', *IEEE Trans. Inf. Forensics Sec.*, 2015, **10**, (3), pp. 507–518

4. Dong, J., Wang, W.: 'CASIA tampered image detection evaluation (TIDE) database, v1.0 and v2.0', http://forensics.idealtest.org/, 2018.

5. Tarman S.H.: 'M-SIFT: A detection algorithm for copy move image forgery'. Proc. of the 4th IEEE int. Conf. on Signal Processing, Computing, and Control. ISPCC2k17, Solan, India, 2019, pp. 425–430.

6. Fengli, , Z, Qinghua, L.: 'Deep learning-based data forgery detection in automatic generation control'. Proc. of the IEEE Conf. On Communications and Network Security (CNS): Int. Workshop on Cyber-Physical Systems Security (CPS-Sec), Las Vegas, NV, USA, 2017, pp. 400– 404

7. Thirunavukkarasu, V., Kumar, J.S.: 'Passive image tamper detection based onfast retina keypoint descriptor'. Proc. of the IEEE Int. Conf. on Advances in Computer Applications (ICACA), Coimbatore, India, 2016, pp. 279–285.

8. Cheng, D., Meng, G.: 'Fusion net edge aware deep convolutional networks for semantic segmentation of remote sensing harbor images', *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 2020, **10**, (12), pp. 5769–5783

9. Li, Y., Liu, X.: 'SIFT keypoint removal and injection via convex relaxation'.

10. Husain, F., Schulz, H., Dellen, B., *et al.*: 'Combining semantic and geometric features for object class segmentation of indoor scenes', *IEEE Robot. Autom.lett.,* 2017,2,(1),pp. 49-55

11. Amerini, I., Ballan, L., Bimbo, A.D.: 'A SIFT-based forensic method for copy–move attack detection and transformation recovery', *IEEE Trans. Inf.Forensics Sec.*2011,6,(3), pp 1099- 1110.

12. Li, J.: 'Active learning for hyperspectral image classification with a stacked autoencoders based neural network'. Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Tokyo, Japan, 2017, pp. 1–4.