Project Report

on

BeClear: Blur Detection in Images Using CNN

Deep Learning

By

Raviteja Kompalli (190330202)

under the supervision of

Mrs. Deepthi Kalavala

Department of Computer Science and Engineering

K L University Hyderabad,

Aziz Nagar, Moinabad Road, Hyderabad – 500 075, Telangana, India.

May, 2022

Declaration

The Project Report entitled "**BeClear: Blur Detection in images using CNN** "is a record of bonafide work of **V. Nitya Santhoshini (190330249), Raviteja Kompalli (190330202)** submitted as a requirement for the completion of the course **Deep Learning** in the Department of Computer Science and Engineering to the K L University, Hyderabad. The results embodied in this report have not been copied from any other Departments/University/Institute.

(V. NITYA SANTHOSHINI)

(RAVITEJA KOMPALLI)

Certificate

This is to certify that the Project Report entitled "**BeClear: Blur Detection in images using CNN**" is being submitted by **V. Nitya Santhoshini (190330249), Raviteja Kompalli (190330202)** as a requirement for the completion of the course **Deep Learning** in the Department of Computer Science and Engineering, K L University, Hyderabad is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/ University/Institute.

Signature of the Supervisor

(**Mrs. Deepthi Kalavala)**

Signature of the HOD

Signature of the Examiner

ACKNOWLEDGEMENT

First and foremost, we thank the lord almighty for all his grace & mercy showered upon us, for completing this project successfully.

We take grateful opportunity to thank our beloved Founder and Chairman who has given constant encouragement during our course and motivated us to do this project. We are grateful to our Principal **Dr. L. Koteswara Rao** who has been constantly bearing the torch for all the curricular activities undertaken by us.

We pay our grateful acknowledgement & sincere thanks to our Head of the Department **Dr. Chiranjeevi Manike** for his exemplary guidance, monitoring and constant encouragement throughout the course of the project. We thank **Mrs. Deepthi Kalavala** of our department who has supported throughout this project holding a position of supervisor.

We whole heartedly thank all the teaching and non-teaching staff of our department without whom we won't have made this project a reality. We would like to extend our sincere thanks especially to our parents, our family members and friends who have supported us to make this project a grand success.

ABSTRACT

With the availability and usage of digital cameras in today's world, the number of digital images increases gradually which increases the demand for automatic image quality assessment. One of the major checks in image quality is detecting blurriness in the image. There are many methods to detect blurriness in the image where some of them are using OpenCV and multi-layer perceptron. These methods give less accuracy models on the dataset which may not be feasible for applying in real-time applications. Hereby, we proposed a solution which uses convolutional neural networks which predict whether the image is blurry or not.

TABLE OF CONTENTS

INTRODUCTION

As the popularity of digital cameras is increasing, many have a huge collection of digital pictures. Even the photographers who take pictures in ceremonies like weddings, birthday parties and anniversary celebrations confess that almost 40% of the images taken by them, have a less quality of pictures. One of the reasons for such poor quality of pictures is blurriness in the images. Therefore, Blur detection on these images helps the photographers or users to classify the images so that one can move all the blurred images into a separate folder so that their efforts in separating out the images becomes less.

Blur detection is more challenging as it is hard to distinguish the blur type, blur level and blur setting . It segments the blurred parts of an image accurately. It is an important application in salient object detection, image restoration, defocus magnification, deblurring, blur segmentation and so on. According to the cause of Blurness,the blur is divided into 2 types i.e., out-focus and motion blur. Blur detection is a baseline for many applications as mentioned above. Usually, Blur is purposefully added by the photographers or the cameraman to add effects to the image. This skill is very common by optical imaging systems. All these hand-crafted feature-based methods are convenient and effective.

There are many methods in the field of machine learning where we can detect blur in a given image and classify the image as blur or clear image. Some of the methods are using variance of Laplacian using OpenCV library in python, using MLPCLassifier of sklearn library, implementation of Convolutional neural networks using TensorFlow and keras libraries in python. These methods give our desired output in a single floating value point which represents the class i.e., blur or clear. These methods differ in giving the accurate results.Though,some methods can give good and appropriate results, other methods can give an inappropriate result while some may give accurate results based upon how we work in each method.

The goal of this project is to build a model for a mobile as well as desktop application which predicts if the input image is blurred or not blurred based on the model which is trained using convolutional neural networks.

TERMINOLOGY

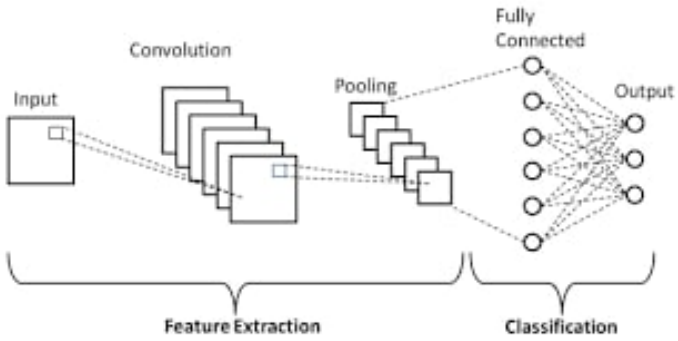
Convolutional Neural Networks: Convolutional neural network is a class of deep learning methods which has become dominant in various computer vision tasks and is attracting interest across a variety of domains, including radiology. A convolutional layer is the main building block of a CNN. It contains a set of filters (or kernels), parameters of which are to be learned throughout the training. The size of the filters is usually smaller than the actual image. Each filter convolves with the image and creates an activation map. The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision. Convolutional neural network is composed of multiple building blocks, such as convolution layers, pooling layers, and fully connected layers, and is designed to automatically and adaptively learn spatial hierarchies of features through a backpropagation algorithm.

OpenCV : OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. OpenCV is a great tool for image processing and performing computer vision tasks. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection, and much more. It supports multiple languages including python, java,C++. Computer Vision can be defined as a discipline that explains how to reconstruct, interrupt, and understand a 3D scene from its 2D images, in terms of the properties of the structure present in the scene. It deals with modeling and replicating human vision using computer software and hardware.

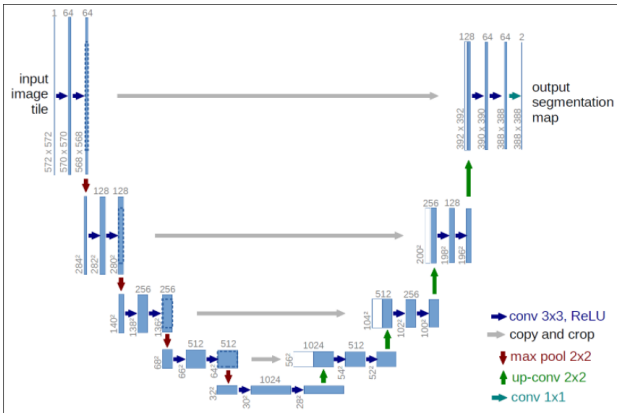
U-Net Architecture : With this U-Net architecture, the segmentation of images of sizes 512X512 can be computed with a modern GPU within small amounts of time. There have been many variants and modifications of this architecture due to its phenomenal success. Some of them include Ladder Net, U-Net with attention, the recurrent and residual convolutional U-Net (R2-UNet), and U-Net with residual blocks or blocks with dense connections. Although U-Net is a significant accomplishment in the field of deep learning, it is equally essential to understand the previous methods that were employed for solving such kind of similar tasks. The U-Net is an elegant architecture that solves most of the occurring issues. It uses the concept of fully convolutional networks for this approach. The intent of the U-Net is to capture both the features of the context as well as the localization. This process is completed successfully by the type of architecture built. The main idea of the implementation is to utilize successive contracting layers, which are immediately followed by the up-sampling operators for achieving higher resolution outputs on the input images.

RELATED WORK

In the beginning, we have used variance of Laplacian using OpenCV where we get the threshold using that we get a blur rate to classify the image. Using this we have got a less accuracy ~62% then using the Sklearn library available, we have used MLP classifier to train our model with that trained model we have got an accuracy of ~78%. And upon analyzing the things we came to know that there are some corner cases such as image of a plan wall, image of a picture taken in portrait mode. These were wrongly classified. Then we have decided to implement Convolutional Neural Networks using Keras and TensorFlow which we found as the simple and easiest technique. It gives the output as a single floating-point value which represents the class of the image i.e., blur or clear.



We have used Convutional neural networks on a pretrained model i.e., Unet for implementing blur detection on the images. The Unet Architecture is shown in figure.
A CNN-trained model based on a dataset which contains training and evaluation directories in which there exists other two directories namely Naturally-blurred, artificially-blurred and undistorted image directories in training set and naturally-blurred and artificially-blurred directories under evaluation set of Certh image blur dataset.



Our dataset uses two types of blurs i.e., natural blur and artificial blur to classify the images. When there is any disturbance in the movement of the camera while clicking the picture, the image obtained is said to be naturally blurred. If the lens of the camera is not focused, then the image obtained is said to be artificially blurred. If the image is clear without any disturbance, then it is said to be undistorted.

As we are giving any kind of images without any specifications, we may get some specific use cases where the system may give an unpredicted answer. Some of those use cases include

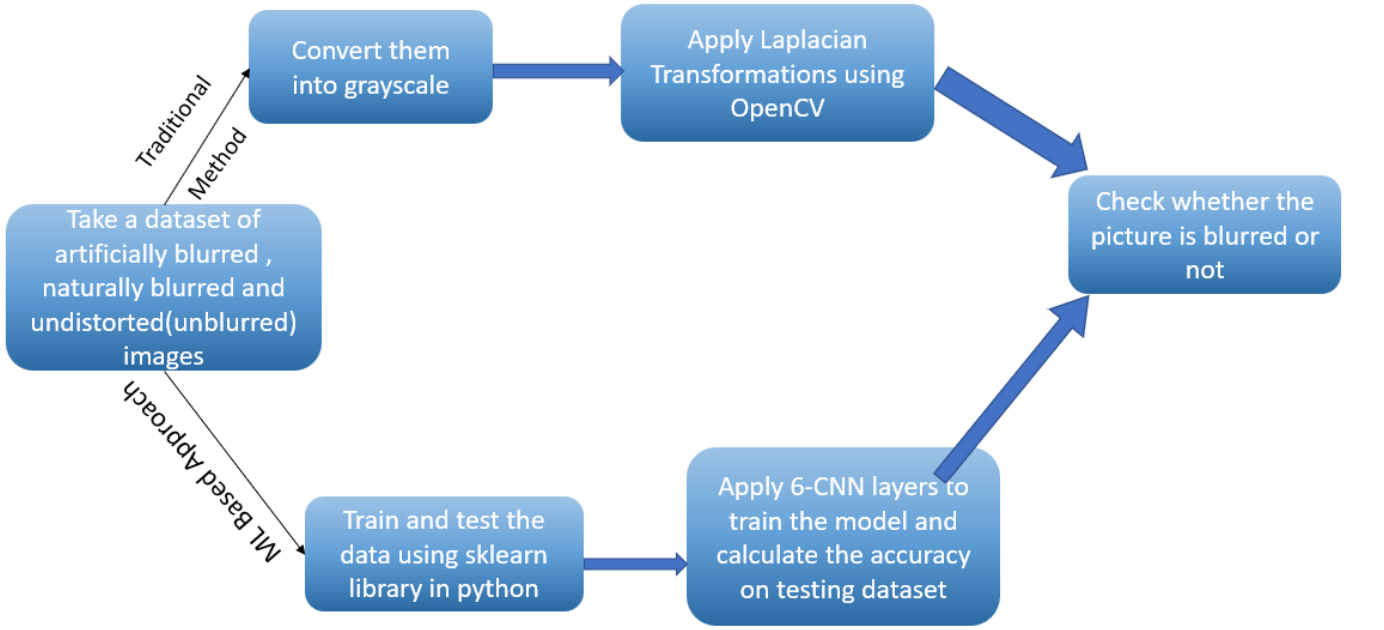
1) An empty wall which should be interpreted as a clear image,
2) An image taken in Portrait mode from an iPhone which should be portraited as a clear image,
3) An image which contains 2 ppl standing diagonally in which a person standing away is clear whereas another person is blurred. This scene should give us a blur verdict,

These are some of the use-cases which are implemented in real life.

OBJECTIVE

The Aim of our project is to implement a model which works for mobile as well as desktop applications where we drop images of different kinds and segregate whether the image is blurry or not. In demand of using digital cameras, there is a huge collection of digital images near every person. Moreover, the photographers who take images in birthday parties, weddings etc.., admit that 30-35% of the images have low quality are sent to a client. One of the key factors that lead to quality degradation of the image is detection of blur in the image.

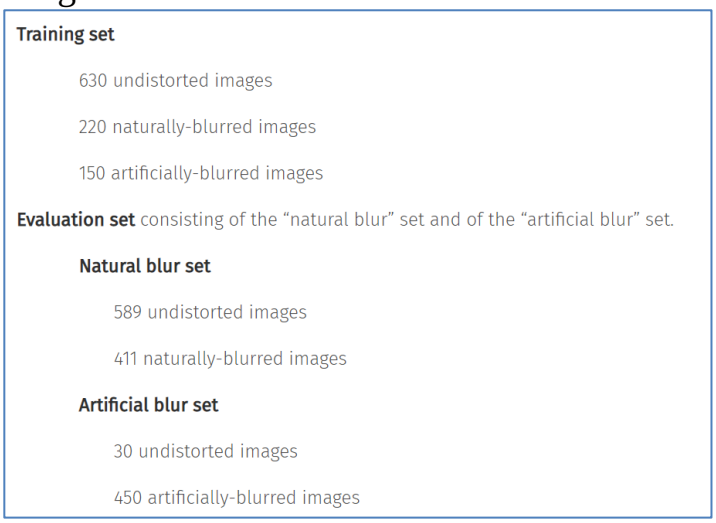
We finally came up, started working work with a real time project which is time efficient where it will be helpful to all the people who face the problem of quality degradation of the image as they can immediately detect when the picture clicked is blurred or not.



REQUIREMENTS AND DATASETS

- **Dataset**: Certh image dataset
- **Processor**: 64-bit, 4 core, 2.5 GHz minimum per core
- **RAM**: 16 GB
- **Hard Disk**: 10 GB for installation

Datasets are taken from http://mklab.iti.gr/files/imageblur/CERTH_ImageBlurDataset.zip It contains training and evaluation directories in which there exists other two directories namely Naturally-blurred, artificially-blurred and undistorted image directories in training set and Naturally blurred and artificially blurred directories under evaluation set of Certh image blur dataset.



MODULES

The different modules used in this project are:

- Load the model
- Upload the image
- Check if the image is blurry or clear

PROPOSED SOLUTION

BeClear is a mobile or a desktop application where an image of any kind is given as an input to the system and it tells us whether the image is blurred or clear. We have trained or model using the below steps.

Process of Implementation :

We used 6 X 6 convolutional neural networks (CNN) to train the dataset and some of the image processing techniques like edge detection and segmentation.
First, import all the directories from the dataset and study the given data.

Then we train the whole data using keras image data generator library to make the data ready for the model.
To train the data we need to do some changes in the image so we do the augmentation of the images.

About the architecture :

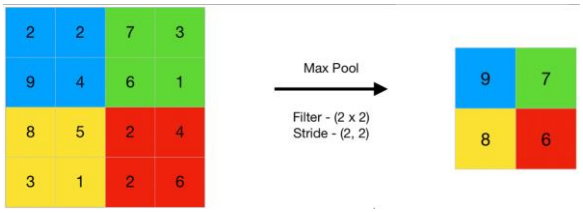
We have used Sequential technique – CNN to train our model.
We have added 3 pooling layers, 3 dense layers, 3 dropout layers, 3 convolution layers, 1 flatten layer and, the activation functions used to train our model is ReLU and Soft Max.
ReLU is used in Hidden Layers and SoftMax is used in the output layer

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 222, 222, 32)      896

 conv2d_1 (Conv2D)           (None, 220, 220, 64)      18496

 max_pooling2d (MaxPooling2D  (None, 110, 110, 64)     0
 )

 dropout (Dropout)           (None, 110, 110, 64)      0

 conv2d_2 (Conv2D)           (None, 108, 108, 64)      36928

 max_pooling2d_1 (MaxPooling  (None, 54, 54, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 54, 54, 64)        0

 conv2d_3 (Conv2D)           (None, 52, 52, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 26, 26, 128)      0
 2D)

 dropout_2 (Dropout)         (None, 26, 26, 128)       0

 flatten (Flatten)           (None, 86528)             0

 dense (Dense)               (None, 64)                5537856

 dropout_3 (Dropout)         (None, 64)                0

 dense_1 (Dense)             (None, 2)                 130

=================================================================
Total params: 5,668,162
Trainable params: 5,668,162
Non-trainable params: 0
_________________________________________________________________
```

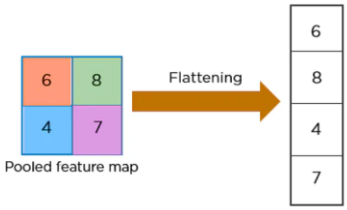
Convolution 2D Layer: This layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs. In this, A bias vector is created and added to the outputs. Finally, if activation is not None, it is applied to the outputs as well.

Maxpooling Layer : Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map.



Dropout Layer : The Dropout layer is a mask that nullifies the contribution of some neurons towards the next layer and leaves unmodified all others. We can apply a Dropout layer to the input vector, in which case it nullifies some of its features; but we can also apply it to a hidden layer, in which case it nullifies some hidden neurons.
Dropout layers are important in training CNNs because they prevent overfitting on the training data.

Flatten Layer : Flattening is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector.



Dense Layer : Dense Layer is simple layer of neurons in which each neuron receives input from all the neurons of previous layer, thus called as dense. Dense Layer is used to classify image based on output from convolutional layers.

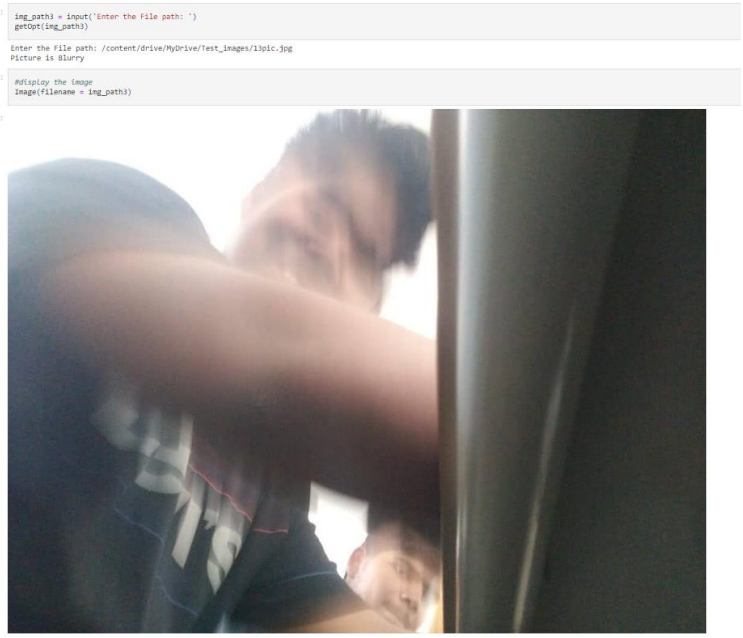
The activation function used were ReLU and SoftMax as the output layer.
We have use ReLU activation function because piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.

Training and validating the model :

Then we filter the data using train generator and test generator and divide them based on their classes i.e., blurred or clear.
Now, we train the model. We have done 100 epochs to train this model and the result was accurate.
Then we have tested the model which was trained and saved the model to run for the mobile or a desktop application.

RESULTS DISCUSSION

We need to provide the input as an image of any format and it predicts whether the image is blurred or clear.
If the model gets implemented in web apps or mobile apps, then it can be directly installed and used. Our idea: one can use this model and segregate out the images based on its class i.e., blurred or clear . with this model one can move all the images that are blurred into a new directory.
The input in the interface can be as follows:



The output in the interface is as follows:

In [ ]:
```
img_path2 = input('Enter the File path: ')
getOpt(img_path2)
```

```
Enter the File path: /content/drive/MyDrive/Test_images/pic_5.jpg
Picture is Blurry
```

In [ ]:
```
#display the image
Image(filename=img_path2)
```

```
img_path2 = input('Enter the File path: ')
getOpt(img_path2)
```

```
Enter the File path: /content/drive/MyDrive/Test_images/pic_2.jpg
Picture is Clear
```

CONCLUSION

Blur detection have become a hot topic in the field of computer vision. There are many methods in the field of machine learning where this application can be implemented. There are also different types of blurs in Computer vision like motion blur, artificial blur, natural blur and many more. So, to handle such kind of blurs we need a model which can work on large dataset and can give a good amount of accuracy. Our paper proposes this kind of model using Unet architecture of state-of-art architecture of CNN which can solve the above problem. It provides the implementation of Unet architecture on Blur detection. The future work may include measuring the amount of blurriness in the blur image so that it could be helpful for the photographers to classify the images based on the amount of blur present in the image.

In future, we would like to extend this project and deblur the picture which is detected as blur and convert it to a clear picture using GAN's in deep learning.

REFERENCES

[1] https://github.com/ILasya/Image-Quality-Detection/blob/main/CNN-Model-Evaluation.ipynb

[2] https://www.youtube.com/watch?v=nHQDDAAzIsI

[3] https://github.com/priyabagaria/Image-Blur-Detection

[4] Tomasz Szandała, Convolutional Neural Network for Blur Images Detection as an Alternative for Laplacian Method,2021.

[5] Xiaoli Sun, Qiwei Wang, Xiujun Zhang, Chen Xu and Weiqiang Zhang, Deep blur detection network with boundary-aware multi-scale features, June 2021.

[6] Kai Zeng , Yaonan Wang, Jianxu Mao, Junyang Liu. Weixing Peng, and Nankai Chen, A Local Metric for Defocus Blur Detection Based on CNN Feature Learning,2018.

[7] Beomseok Kim, Hyeongseok Son, Seong-Jin Park , Sunghyun Cho , and Seungyong Lee, Defocus and Motion Blur Detection with Deep Contextual Features,2018.

[8] Caglar Senaras, M. Khalid Khan Niazi, Gerard Lozanski, Metin N. Gurcan, DeepFocus: Detection of out-of-focus regions in whole slide digital images using deep learning, October 2018.