International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 03 | Mar 2020 www.irjet.net p-ISSN: 2395-0072

Single Image Super Resolution Using Machine Learning

Patel Safiyabegum[1], Kulkarni Nikita[2], Galate Alfa[3], Sabungar Anzalnagohar[4], Khan Asiya[5]

[1,2,3,4]Student, Dept of Computer Science & Engineering, BMIT, Solapur, Maharashtra, India
[5]Professor, BMIT college of Engineering, Solapur, Maharashtra, India

---

Abstract - *High resolution (HR) image reconstruction from single low-resolution (LR) image is one of the most important vision applications. Despite numerous algorithms have been successfully proposed in recent years, efficient and robust single image super resolution (SISR) reconstruction is still challenging by several factors, such as inherent ambiguous mapping between the HR and LR images, necessary use exemplar images, and computational load. This paper presents the single image super resolution algorithm which mainly focuses on constructing a high quality and high-resolution image from a low resolution, low quality image. It is useful in many cases like medical imaging, satellite imaging, surveillance, crime investigation and video applications etc.*

Key Words: **Super Resolution, Image Reconstruction, Single Image Resolution Techniques, Resolution enhancement.**

I. INTRODUCTION

This project focuses on image super resolution to enhance the quality of images by taking one low resolution image as input and producing a high-quality image as a output. The main advantage of using this algorithm is cost efficiency. It is beneficial in areas like satellite produced images, surveillance images from borders, crime investigation and in disease diagnose. The system uses FANN (Fastest Artificial Neural Network) to map the resolutions of image. Before mapping the system it retrieves the values of pixels, PSNR(Peak Signal-To-Noise Ratio) and mean square error(MSE) in image and trains itself to reduce the noise and errors from image and performs down sampling, up scaling on image and generates datasets and stores in files color-wise and then the datasets are given to neural network to process and this process continues till it gets the satisfactory results and produces a high quality image to the user .

II. LITERATURE REVIEW

This system accepts low resolution image (i.e. noisy image, blur parts etc.), system starts to capture the details of image. The system itself makes multiple images which captures different details of image and tries to find out patterns, quality, contrast of image etc. The system takes small blocks of image (16 * 16) and then it calculates noise ratio and generates data sets those data sets are passed to neural network and processed and generates a bicubic image from blocks of input image. During this process actual loss of bits in values is calculated. Once done with this stage the bicubic image is given as input to ANN which in turn does the same processing and generates a high-quality image.

This paper represents a fastest artificial neural network algorithm which works so efficiently and gives maximum throughput in less amount of time. The algorithm automatically finds outs the edges corners, color contrast, blurry part , noisy part from image . While performing image generation in high quality neural network creates a blank image and transfers data calculated on data sets generated and fill the blocks using nearest neighbor technique. This whole system is divided into two parts namely train model and then test model while training model has many small modules.

1. Train model

Nowadays technology has emerged and grown a lot, having high quality images has become necessity especially in areas like defense, security, medical and crime investigation departments having highly configured systems in day to day life even for general systems (public places, stores, shopping malls) is difficult as it becomes expensive and requires high maintenance. This project overcomes this drawback and provides high resolution images in less time.

To train model the first task is to take low resolution images and high-resolution images, after this step the next step is to retrieve pixel values color wise and stores in 3 different files (datasets). After these two steps next step is to create a neural network by specifying number of input layers, hidden layers and output layers, learning rate after these datasets are fed into the neural network and model starts to train learns the difference between high and low resolution images. Apart from this it detects the different edges of an image and uses different methods and generates bicubic image and does same processing on bicubic image and learns aspects of image. This system can take images of any object like human faces, animals, nature, image from a video like from CCTV footage etc.

2. Test model

Second main part is to test the system. While testing on an image user can give any low-resolution image as input and user will get a high-resolution image as output. It gives satisfactory results.

III. METHODOLOGY

i. FILTERING

At the beginning, images has to be inputted into the system and it creates 16 x 16 blocks of image to capture each and

every minor detail of image and simultaneously system resizes those 16 x 16 blocks into 8 x 8 blocks to learn the difference of pixel value finds PSNR(Peak signal to noise ratio) values and store pixel values into files that files are called datasets.

Datasets are stored in RGB color format, that generated datasets are the main root of the project as it uses supervised type learning data is labeled and classifying image values color-wise makes system work faster. These generated datasets are passed to ANN (artificial neural network).

ii. BICUBIC INTERPOLATION

Image interpolation is the process of converting the image from one resolution to other resolution. This process is performed on a one-dimension basis row by row and then column by column. Image interpolation estimates the intermediate pixel between the known pixels by using different interpolation kernel. Bi-cubic Interpolation In cubic interpolation intensity at point is estimated from the intensity of 16 closest to it. The basis function is Bi-cubic gives smooth image but computationally demanding. Following fig (1) shows the general working of bicubic interpolation.

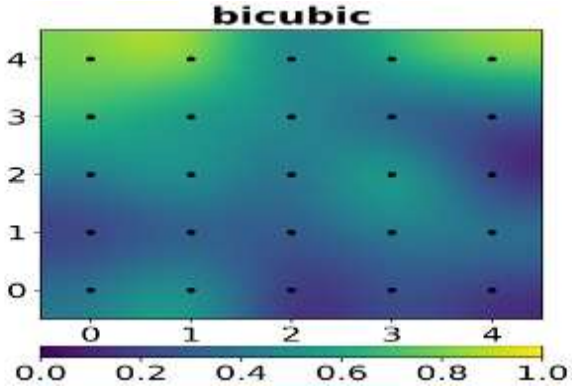

Fig (1): Bicubic interpolation

iii. SEQUENTIAL MODEL

The simplest model is defined in the Sequential class which is a linear stack of Layers. You can create a Sequential model and define all the layers in the constructor, for example:

from keras.models import Sequential

model = Sequential(...)

A more useful idiom is to create a Sequential model and add your layers in the order of the computation you wish to perform, for example:

from keras.models import Sequential

model = Sequential()

model.add(...)

model.add(...)

model.add(...)

Model Inputs

The first layer in your model must specify the shape of the input.

This is the number of input attributes and is defined by the input_dim argument. This argument expects an integer.

Dense(16 ,input_dim = 8)

Model Layers

Layers of different type are a few properties in common, specifically their method of weight initialization and activation functions.

1) Weight Initialization

The type of initialization used for a layer is specified in the init argument. Some common types of layer initialization include:

- "*uniform*": Weights are initialized to small uniformly random values between 0 and 0.05.
- "*normal*": Weights are initialized to small Gaussian random values (zero mean and standard deviation of 0.05).
- "*zero*": All weights are set to zero values.

  You can see a full list of initialization techniques supported on the Usage of initializations page.

2) Activation Function

Keras supports a range of standard neuron activation function, such as: softmax, rectifier, tanh and sigmoid.

You typically specify the type of activation function used by a layer in the activation argument, which takes a string value.

You can see a full list of activation functions supported by Keras on the Usage of activations page.

Interestingly, you can also create an Activation object and add it directly to your model after your layer to apply that activation to the output of the Layer.

3) Layer Types

There are large number of core Layer types for standard neural networks. Some common and useful layer types you can choose from are:

- **Dense**: Fully connected layer and the most common type of layer used on multi-layer perceptron models.
- **Dropout**: Apply dropout to the model, setting a fraction of inputs to zero to reduce over fitting.
- **Merge**: Combine the inputs from multiple models into a single model.
- You can learn about the full list of core Keras layers on the Core Layers page

Model Compilation

Once you have defined your model, it needs to be compiled.

This creates the efficient structures used by the underlying backend (Theano or TensorFlow) in order to efficiently execute your model during training.

You compile your model using the compile() function and it accepts three important attributes:

1. Model optimizer.
2. Loss function.
3. Metrics.

model.compile(optimizer=, loss=, metrics=)

Model Prediction

Once you have trained your model, you can use it to make predictions on test data or new data.

There are number of different output types you can calculate from your trained model, each calculated using a different function call on your model object. For example:

- *model.evaluate()*: To calculate the loss values for input data.
- *model.predict()*: To generate network output for input data.
- *model.predict_classes()*: To generate class outputs for input data.
- *model.predict_proba()*: To generate class probabilities for input data

Summarize the Model

Once you are happy with your model you can finalize it. You may wish to output a summary of your model. For example, you can display a summary of a model by calling the summary function, for example :

Model.summary()

IV. RESULT

V. CONCLUSIONS

1. We have constructed a deep neural network to process and enhance quality of images .The major advantage of the super resolution approach is that it may cost less and the existing LR imaging systems can be still utilized.
2. Synthetic zooming of region of interest (ROI) is another important application surveillance, forensic, scientific, medical, and satellite imaging.
3. This application is most suitable for magnifying

   objects in the scene such as the face of a criminal or the license plate of a car.

VI. ACKNOWLEDGEMENT

This research was supported by BMIT college of engineering Solapur. We are thankful to our guide

Prof. Asiya Khan who provided her expertise in Machine Learning area that greatly assisted to perform the research.

VII. REFERENCES

[1] https://ieeexplore.ieee.org/document/7802119
[2] https://ieeexplore.ieee.org/document/5459271
[3] https://medium.com/@birla.deepak26/single-image-super-resolution-using-gans-keras-aca310f33112
[4] https://medium.com/@jonathan_hui/gan-super-resolution-gan-srgan-b471da7270ec