Table of Contents

1 INTRODUCTION

1.1 Background

Basically, blurring means the loss of contrast and sharpness of an image. The solution for this problem is the image restoration techniques such as image de-blurring. Image de-blurring is the procedure that tries to reduce the blur from the degraded set of images.

When the world is getting smarter with the advent of new and ever developing technology, many consequences are now on surface. These consequences encompass many direct and indirect impacts of smartphones in the lives of the billions. Photographs are now being taken on scale like never before. Photographs are not only the selfies or wefies we take. Photographs are also strong proofs of certain events, certain tasks or even some misdoings. Photographs can capture crimes. Photographs can capture someone's achievement. As it is said "A picture is worth a thousand words", de-blurring a picture can mean discovering a hidden pool of truths. When it comes to taking a good photograph, it's necessary for the photographer to be still, camera needs to be of good quality and many other factors influence the quality of photo. And, de-blurring a blur image would help solve many mysteries that lie inside those m * n dimensional matrices of RGB values or gray scales.

Shaking of hands during a photo session or movement by the one being captured are reasons for blurring of images. This is relative motion between the object and camera is termed as motion blur. Another reason for blur is noise, which can be due to sensor noise or turbulence in air. Camera misfocus is yet another reason for blurring of images. Our primary goal is to develop two models to solve the problem of deblurring. For that, the first model will be dependent completely on the classical approach. This classical approach involves different mathematical operations which includes different digital image processing techniques like deconvolution directly on the bare pixel values of the images. The other model will be the development and implementation of the fancier system of Neural Network. This neural network will be basically a Convolutional Neural Network, CNN in short, whose credit goes to Facebook's head reasearch in ML and Computer Vision, Yann LeCunn.

1.2 Objectives

Our primary objective is to develop a system that implements the state of art technique called Neural Network and classical branch of Mathematics, Digital Singal Processing to deblur a blurred image.

Our objectives can be pointed as:

- To develop a system that implements Neural Network (ML model) which is capable of deblurring a blur image.
- To implement the principles and techniques of digital signal processing (DSP model).
- To develop a web interface that can deblur images using both ML and DSP model.

1.3 Scope of the work

Image deblurring is not a very new problem. Our project to target all sorts of blurs which include motion blur, noise blur. The project will work on both blind and non blind deblurring mechanisms. This deblurring can become handy to sort out decreased quality of photos which in turn can be a source of many concrete proofs and truths.

2 LITERATURE REVIEW

2.1 Classical Approach

In this approach, the blur on the image is assumed to be a linear distortion. Using this assumption, a linear equation is formulated that relates the blurred image to the unblurred one. The blurred image is then decomposed using a technique called SVD (Singular Valued Decomposition). The property of the singular valued decomposition is that it allows the error introduced in the image (the blur) to be approximately separated into constituent "frequency" like forms. The blurred image is then the sum of the original image and the errors of the multiple frequencies. The high frequency components of the blur have a high contribution to the blur, so they are discarded and a truncated function is found which depends upon a number $k<N$ where $N = mn$ is the

product of the matrix dimensions representing the image. Adjusting the value of k, various degree of deblurring can be achieved.

The simple linear model is not sufficient to produce a high quality of deblur. In order to further remove the erroneous signals from the image, each pixel of the image is assumed to be a point source of light and different kinds of blurs arising from physical and optical disturbances are modeled by functions of the point source's intensity around the area of the point source. These functions are called PSFs (Point Source Functions). The PSFs are used to construct PSF arrays (blurred images of a single point source of light). We improve the model with the inclusion of the contribution of the PSF blur components in our error equation.

Furthermore, the blurred image is usually an effect of both blur and noise. Different kinds of noise models can be added to the simple linear model [] such as the Poisson noise (caused by background photons), the Gaussian white noise due to random disturbances in measurement and quantization error due to analog-digital conversion.

With the inclusion of both blur and noise, we can evaluate the unblurred image pixel-by-pixel by solving for the obtained linear equation. We evaluate the value of a certain pixel as the blurred value minus the weighted sum of effect of blur and noise from surrounding pixels.

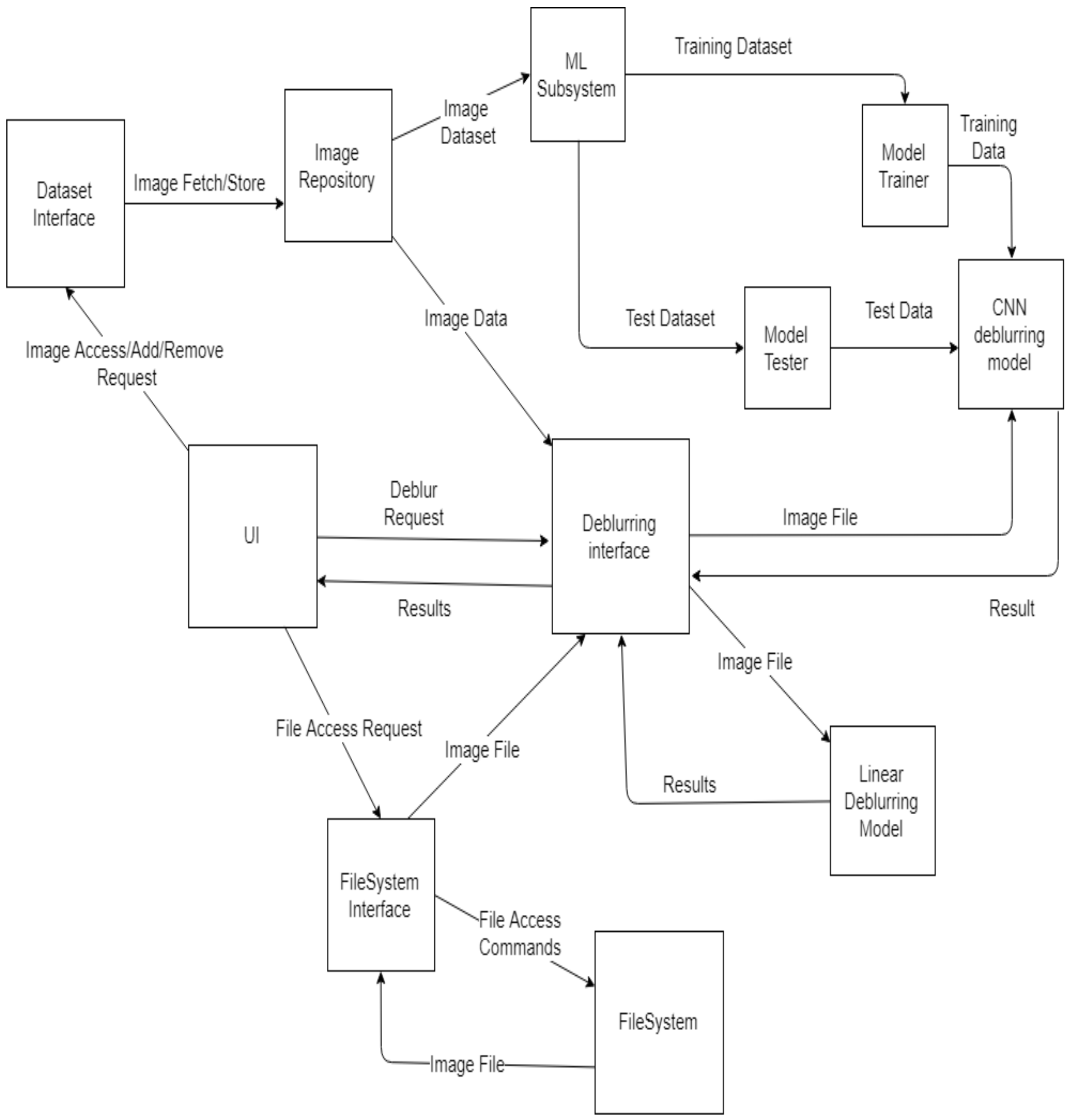
A problem arises, however in the boundary of the image where there is not enough information available to solve the equation. Some information about the blur is lost beyond the boundary. In order to evaluate the values of the pixels at the boundary, we assume different boundary conditions such as zero boundary (assuming everything outside the boundary is black), periodic boundary (assuming everything outside the boundary is periodic repetitions of the image inside), or reflexive boundary (assuming everything outside the boundary is a mirror of the image inside the boundary). Using these boundary conditions, we evaluate the weighted convolution sum of the PSF arrays with the flattened unblurred image to the blurred image. Solving this system yields the actual unblurred image.

2.2 Neural Network Approach

Image deblurring with aid of a Convolutional Neural Network is not a completely new work. Various computer scientists have worked on this discipline of Image Processing. Some notable works include by Reshma Vijay V.J., Deepa P.L. They generated a new CNN that performs the entire deblurring process. That model is not required to perform deconvolution and all. The CNN itself performs every single operation within its hidden layers and gives out a clear output image. Deconvolution algorithms seek to stably invert the linear convolution operation when the blur kernel is known. Blind deconvolution concerns the more challenging case when the kernel is unknown. Andy Gilbert, Shai Messingher and Anirudh Patel investigated whether a learning based approach to non-blind image deconvolution could enhance conventional deconvolution techniques. The hypothesis was that a non-linear combination of conventional reconstructions of the same blurry image yields a sharper image. The system proposed takes in a blurry picture, forms 15 deblurred versions of it using Wiener filtering with different SNR assumptions, stacks them into a tensor, puts them through a deep neural network to non-linearly combine them, and outputs a new reconstructed version.

3 METHODOLOGY

3.1 System Block Diagram



3.2 Description of System

3.2.1 Image deblurring using a linear approximation

In order to understand how to deblur an image, one must understand what causes a blur in the first place. In any computer system, an image is represented by a matrix of numbers. The numbers represent the intensity and the color of the pixels of the image.

For simplicity, we consider at first, the blurring of a grayscale image. The representation of a grayscale image is a matrix X of dimension m x n in which each element X_{ij} is a number from 0 to 2^n (8 for instance) which represents the relative intensity with reference to 0 being perfectly black and 2^n being perfectly white.

Again, consider that the blurred form of this image is represented by a m x n matrix B.

With these two matrices in mind, we first assume that the blurring and deblurring are linear transformations and linear algebra can be used to solve the problem of deblurring. The assumption is justified in most cases.[]

In order to simplify our calculations, we further convert the m x n matrices B and X into matrices b and x respectively, each of dimensions mn x 1. The transformation is done by stacking the columns of the former matrices to form the latter matrices.

Using our assumption of a linear blurring model, we can write

$$\mathbf{A}\,\mathbf{x} = \mathbf{b}, \tag{3.1}$$

Where A is a square matrix of dimensions mn x mn.

We might be under the naïve assumption that such a simple model can be easily solved to produce a good deblur. Such is not the case due to errors in the simplified model owing to mechanical or optical fluctuations (focus, aperture etc.) The quantization error in digitization of the image also plays a part in introducing an error.

Let the error matrix be E_{mxn} and let its vector form (mn x 1) be e.

Then, we can write

$$\mathbf{b} = \mathbf{b}_{\text{exact}} + \mathbf{e} \tag{3.2}$$

Using this equation and solving for x, we get the naïve solution,

$$\mathbf{x}_{\text{naïve}} = \mathbf{A}^{-1}\mathbf{b} = \mathbf{A}^{-1}\mathbf{b}_{\text{exact}} + \mathbf{A}^{-1}\mathbf{e} = \mathbf{x} + \mathbf{A}^{-1}\mathbf{e}, \tag{3.3}$$

In the above equation, the term $A^{-1}e$ is called the inverted noise.

3.2.1.1 Analyzing inverted noise using Singular Value Decomposition (SVD):

The SVD of a square matrix A_{nxn} is a defined as the decomposition

$$\mathbf{A} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T \tag{3.4}$$

Where U and V are orthogonal matrices of dimension N x 1 and 1 x N respectively satisfying $UU^T = VV^T = I_N$ and $\Sigma = \text{diag}(\sigma_i)$ is a diagonal matrix whose elements are in non-decreasing order.

Using these relations, it can be seen that the inverse of A is computed as:

$$\mathbf{A}^{-1} = \mathbf{V}\,\mathbf{\Sigma}^{-1}\mathbf{U}^T \tag{3.5}$$

The matrices A and A^{-1} can be represented as:

$$\begin{aligned}
\mathbf{A} &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \\
&= \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_N \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_N \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_N^T \end{bmatrix} \\
&= \mathbf{u}_1\sigma_1\mathbf{v}_1^T + \cdots + \mathbf{u}_N\sigma_N\mathbf{v}_N^T \\
&= \sum_{i=1}^{N} \sigma_i \mathbf{u}_i\, \mathbf{v}_i^T .
\end{aligned}$$

Similarly,

$$\mathbf{A}^{-1} = \sum_{i=1}^{N} \frac{1}{\sigma_i} \mathbf{v}_i\, \mathbf{u}_i^T .$$

It follows from this relation that:

$$\mathbf{x}_{\text{naïve}} = \mathbf{A}^{-1}\mathbf{b} = \mathbf{V}\,\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{b} = \sum_{i=1}^{N} \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\,\mathbf{v}_i \tag{3.6}$$

and the inverted noise contribution to the solution is given by,

$$\mathbf{A}^{-1}\mathbf{e} = \mathbf{V}\,\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{e} = \sum_{i=1}^{N} \frac{\mathbf{u}_i^T\mathbf{e}}{\sigma_i}\,\mathbf{v}_i \tag{3.7}$$

The following properties generally hold for image deblurring:

- The error components $|u_i^T e|$ are small and are typically of the same order of magnitude for all i.
- The singular values (the diagonal elements of $\sum$) decay to a value of 0. Consequently, the condition number

$$\text{cond}(A) = \sigma_1/\,\sigma_N \tag{3.8}$$

is very large, indicating that the solution is very sensitive to perturbation and rounding off errors.

- The singular vectors corresponding to the smaller singular values typically represent high-frequency information: i.e. as i increases, the vectors u_i and v_i tend to have more sign changes.

As a consequence of the last property, the SVD provides us basis vectors v_i (the blur vector) for an expansion where each basis vector represents a certain “frequency”, approximated by the number of times the entries in the vector changes sign.

So, leaving the high frequency components out, we get a stable truncated expansion term for some k<N:

$$\mathbf{x}_k = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\,\mathbf{v}_i \equiv \mathbf{A}_k^{\dagger}\mathbf{b} \tag{3.9}$$

Where,

$$\mathbf{A}_k^{\dagger} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_k \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_k^T \end{bmatrix} = \sum_{i=1}^{k} \frac{1}{\sigma_i}\mathbf{v}_i\mathbf{u}_i^T \tag{3.10}$$

Using various values of k (k<N) we can get different grades of deblurring.

3.2.1.2 Introducing different blur models to the simple linear model

Additionally, optimizations can be done through further approximation of blur using various kinds of Point Source Functions (PSFs) such as horizontal motion blur, out-of-focus blur, atmospheric turbulence blur and Mofatt blurr as depicted below for the point light source.

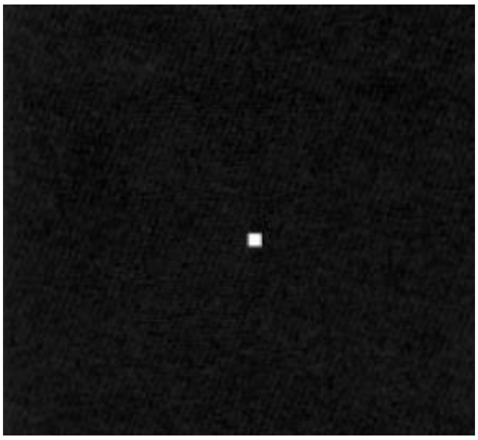


Figure 3.2

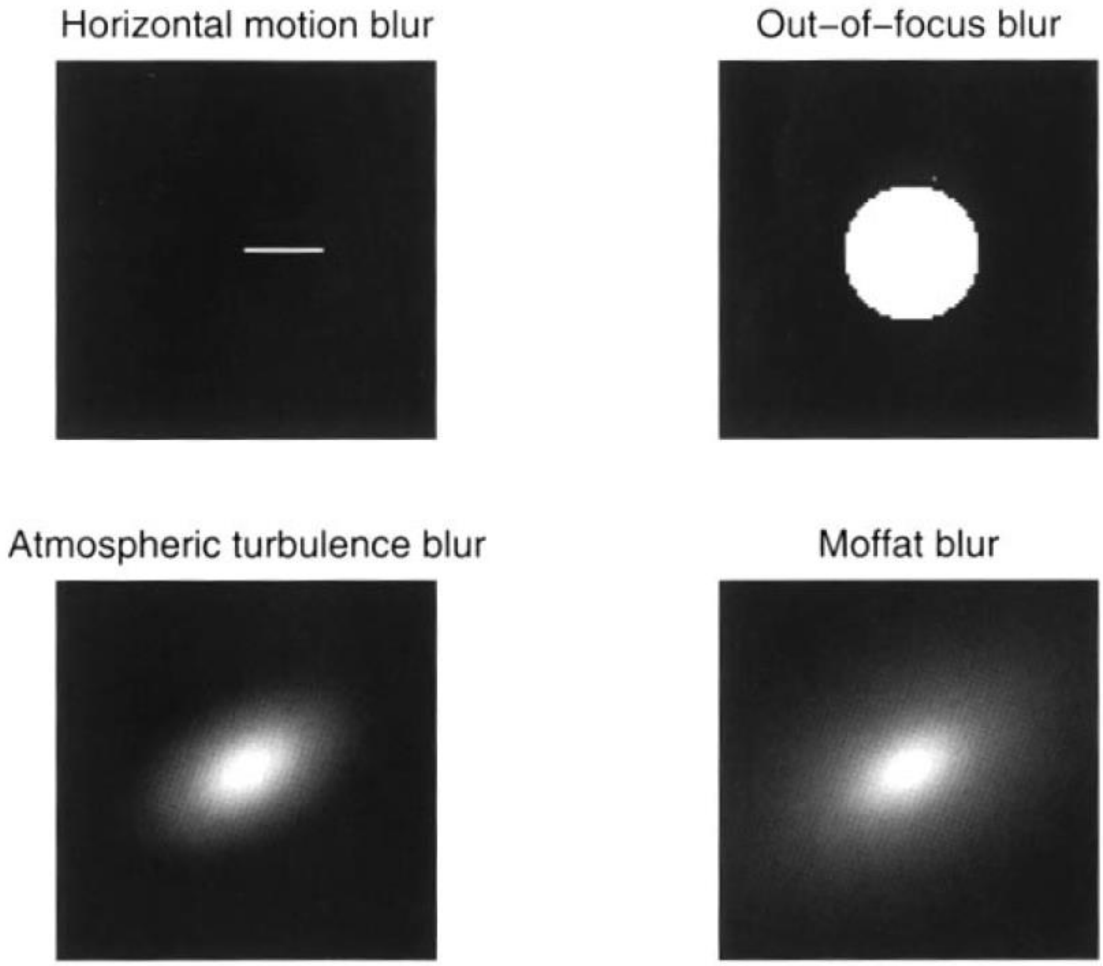


Figure 3.3

The formulation of the PSFs can be derived from the physical motions that cause the blur. For example, the PSF function for the out-of-focus blur are given by:

$$p_{ij} = \begin{cases} 1/(\pi r^2) & \text{if } (i-k)^2 + (j-\ell)^2 \leq r^2, \\ 0 & \text{elsewhere}, \end{cases} \tag{3.11}$$

From the PSF function, a PSF array [p_{ij}] is constructed. Then the blurred image is modeled as a 2 dimensional weighted convolution sum of the unblurred image and the PSF array.

If we need to compute the blurred image B given the sharp image X and the blurring matrix A (created from the PSF arrays), we use:

$$b_i = \mathbf{e}_i^T \mathbf{b} = \mathbf{e}_i^T \mathbf{A}\,\mathbf{x} = \mathbf{A}(i, :)\,\mathbf{x}.$$

(3.12)

Thus, each pixel in the blurred image can be computed as a weighted sum of the corresponding element and its neighbors in the sharp image.

3.2.1.3 The effect of noise

In addition to blur, images are subjected to various kinds of noise. We consider noise from three sources:

- Poisson noise from background photons in the CCD come from both artificial and natural sources.
- Gaussian white noise due to random errors in measurement.
- The noise due to quantization error.

The Poisson noise and the Gaussian white noise components can be drawn from the corresponding probability distributions with a predefined standard deviation. In most cases, (especially with high definition images) the noise due to quantization error can be ignored.

3.2.1.4 Boundary Conditions

Equation 3.12 reveals that the intensity of the pixel in the boundary of the blurred image depends on its neighbor pixels which includes pixels outside the boundary. However, there is no way for us to obtain this information. Without this information, the only way to reconstruct the image is by making educated assumptions about the pixels outside the boundary. We use the following assumptions as referred by.

Zero boundary condition:

With this assumption, we assume that the pixels outside the boundary are perfectly black, so for any pixel value outside the boundary, we use 0 instead. This condition can be represented by embedding our image X in a larger image as:

$$\mathbf{X}_{\text{ext}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{3.13}$$

Periodic boundary condition:

With this assumption, we assume that the entire universe outside our image is a perfect tile of our image. This condition is represented by embedding our image X in a larger image as:

$$\mathbf{X}_{\text{ext}} = \begin{bmatrix} \mathbf{X} & \mathbf{X} & \mathbf{X} \\ \mathbf{X} & \mathbf{X} & \mathbf{X} \\ \mathbf{X} & \mathbf{X} & \mathbf{X} \end{bmatrix} \tag{3.14}$$

Reflexive boundary condition:

With this assumption, we assume that the entire universe outside our image is a reflection of our image. This condition is represented by embedding our image X in a larger image as:

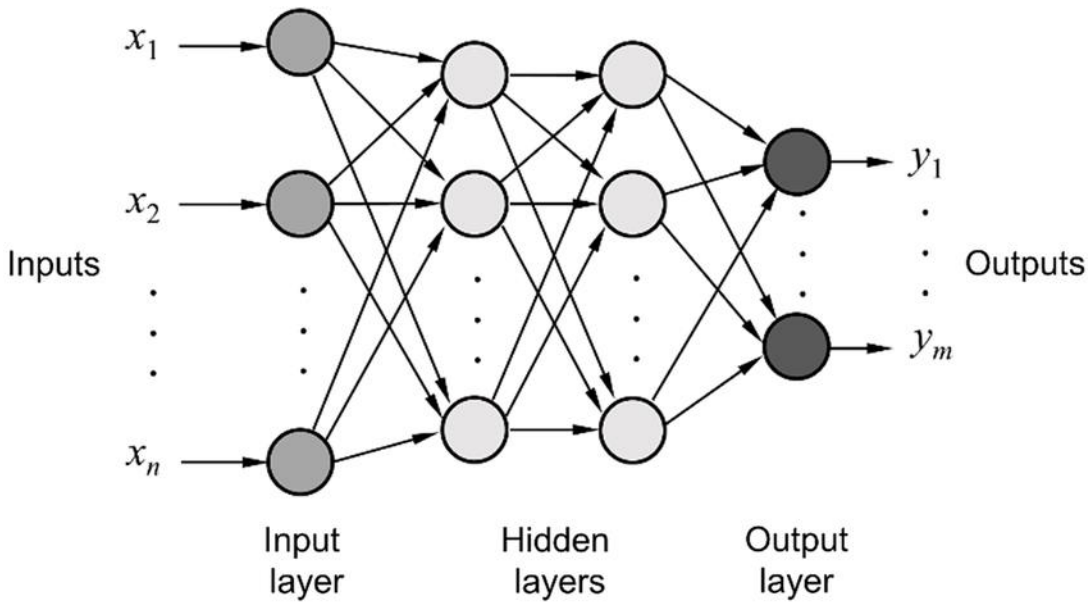
$$\mathbf{X}_{\text{ext}} = \begin{bmatrix} \mathbf{X}_{\times} & \mathbf{X}_{\text{ud}} & \mathbf{X}_{\times} \\ \mathbf{X}_{\text{lr}} & \mathbf{X} & \mathbf{X}_{\text{lr}} \\ \mathbf{X}_{\times} & \mathbf{X}_{\text{ud}} & \mathbf{X}_{\times} \end{bmatrix} \tag{3.15}$$

3.2.2 Image de-blurring using a CNN

Convolutional Neural Networks have been a boon to the field of image processing. It has changed the way for image analysis which used to base only on traditional thinking involving a bunch of complicated mathematics. It’s one of the reasons the field of image processing has grown so far. Handwriting recognition, face detection, etc. are some otherwise difficult problems, which CNN solve with a great accuracy. The unique thing about CNN is that it expects an image as input.

Regular neural networks are magical computation engines. They can estimate the function which maps inputs to outputs to some level of accuracy given a bunch of training sets. Neural Network falls under one of the most important and highly used Machine Learning algorithms. A neural network is a replication of human mind to some extent. It consists of perceptron, which resemble the neurons. A perceptron is designed to fire only when the input is higher than some level. A perceptron has an activation function which gives the final output. Some activation functions include ReLU (Rectified Linear Unit), Sigmoid function and so on. Different factors are considered into understanding of neural network. It has an error function which is to be minimized. These errors can be Mean Square Error, Cross Entropy Error. Gradient descent is one of the techniques that can be implemented to solve this error. In fact many Machine Learning problems fall under the category of cost minimization problem subjected to some constraints. Gradient descent can be applied in different ways and a variety of them include stochastic gradient descent, batch gradient descent. The weight vector that is learnt in order to minimize the error of output being produced and expected output, can be obtained analytically however it comes with a great computational inefficiency. Hence, gradient descent is preferred over analytical solutions. Neural networks may not always give optimal solutions. We try to make curves convex so the final output is optimal. Neural networks can be designed to give either classification outputs or regression outputs. They have a wide variety of accuracy measures. Likewise, for a neural network to act perfectly, it has to have a bias- variance trade-off. Thus simply building a neural network is itself a huge challenge.

Deep neural networks as shown in the following figure implies back propagation in training. Datasets are divided into train, validation and test sets. Different cross validation schemes are used in the process of developing a neural network. A deep neural network has an input layer, output layer and a number of hidden layers.



CNN has a slight variation in the structure than regular neural networks. They are targeted to image processing. CNN is a sequence of layers, and every layer of a CNN transforms one volume of activations to another through a differentiable function. We use three main types of layers to build architectures: **Convolutional Layer**, **Pooling Layer**, and **Fully-Connected Layer** (exactly as seen in regular Neural Networks). We will stack these layers to form a full CNN **architecture**.

Convolutional layer forms convolution in the image and ultimately finds outs patterns with in the image.Max pooling layer reduces the dimensionality of the input image.

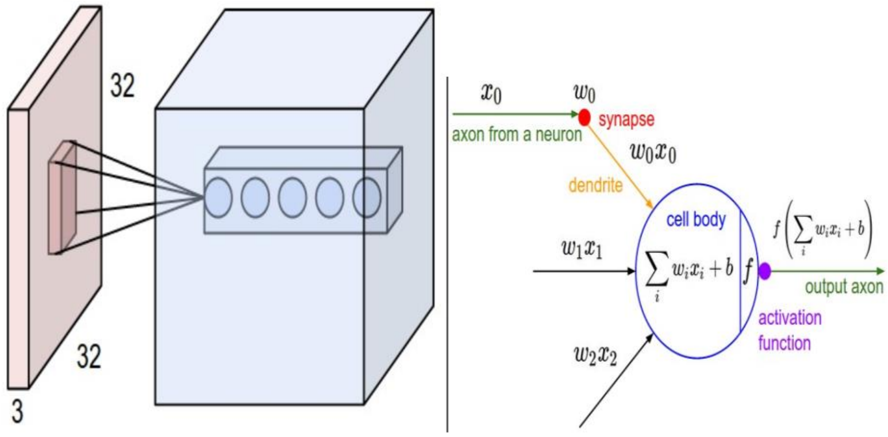


Fig: Convolutional Layer

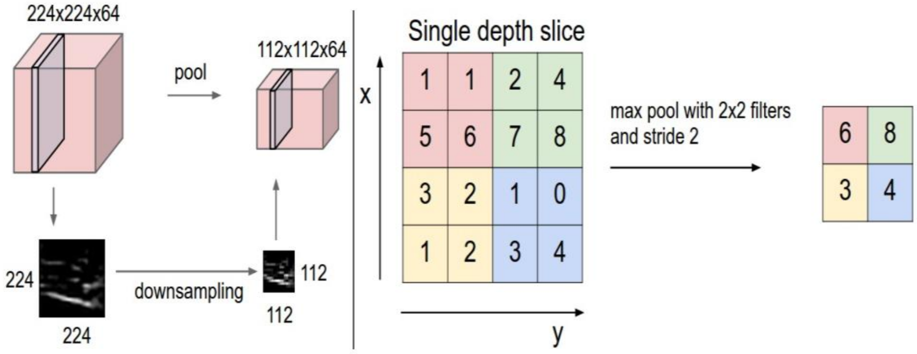


Fig: Max Pooling Layer

The dataset for the application will be developed by adding random noises and motion blur kernels using Tensorflow on fly. And they will be fed to the feed forward model. The model will be made using Tensorflow in Python Programming Language.

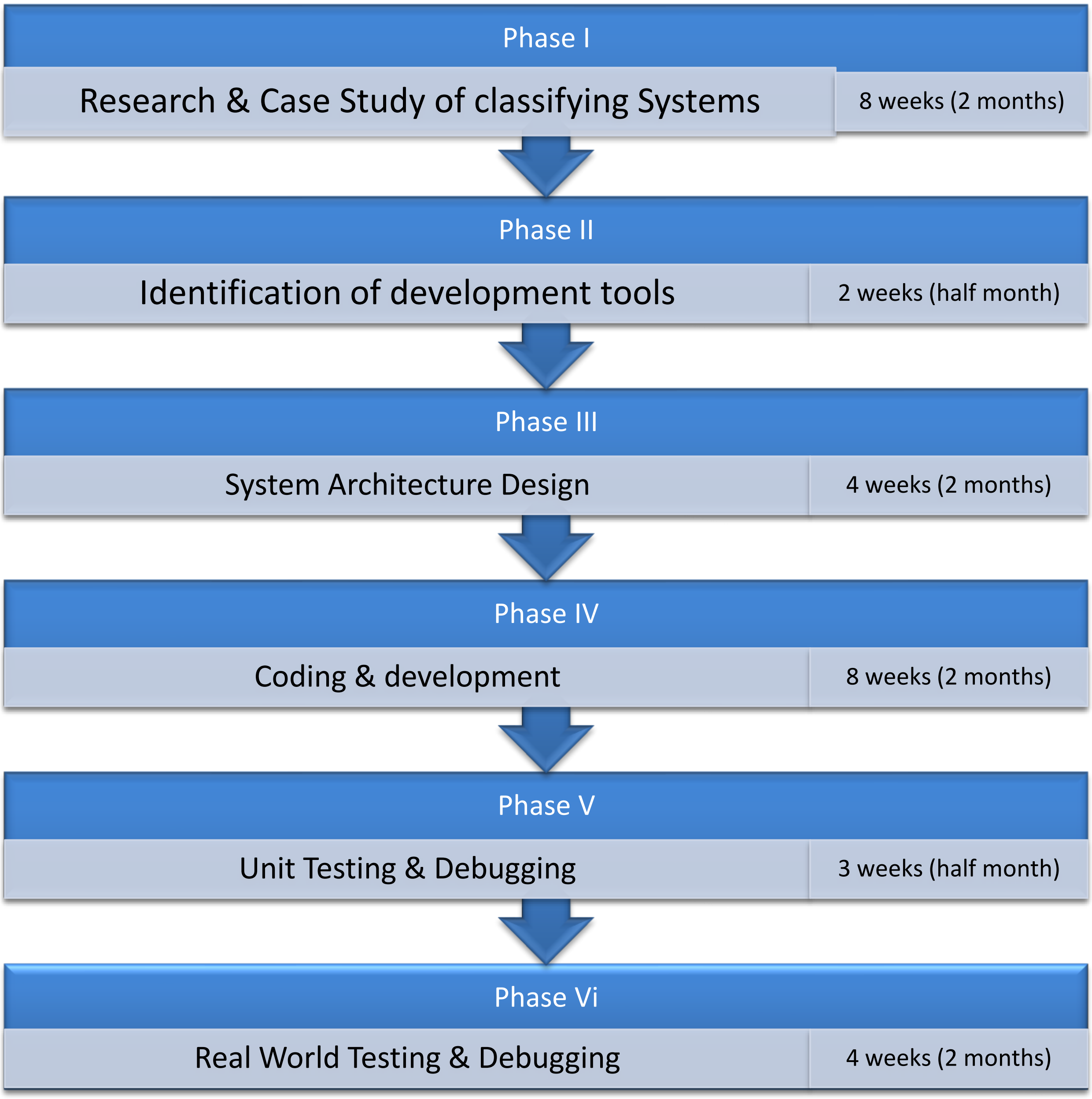
3.3 Web App

The web app will be made to provide interface to the user to use the models. This will be developed using Node.JS in backend, React in Frontend using the models implemented in Tensorflow.

4. EXPECTED OUTCOME

The system will be a web application that can intake blur images and will give back images tuned by our deblurring models. There will be two deblurred images one by the classical approach and another by using neural network. The user can choose which one of the two images is the better one for the case. If the user prefers the classical model’s approach, the image will be a training data for the neural network model and hence the neural network will be trained with that additional data. The user will have options to download the photo or to directly save to Google Drive.

5. PROJECT PLAN



6. REQUIREMENTS

1. Software requirements:

   1.1. Programming Languages: Python, JavaScript, Java

   1.2. Libraries / Frameworks:

- Computation Model: Tensorflow, Pytorch, Numpy, Matplotlib, Scipy, Scikit-learn, OpenCV
- Web App: NodeJs, React framework

2. Resource Requirements:
   2.1. GPU powered Laptops
   2.2. Google CoLab
   2.3. High speed Internet services
   2.4. Past reports on research in ML
   2.5. Past reports on Digital Signal Processing

7. BIBLIOGRAPHY

[1]. Aittala M., Durand F.: Image Deblurring Using Convolutional Neural Network, Reshma Vijay V.J., Deepa P.L (2016)

[2]. Burst Image Deblurring Using Permutation Invariant Convolutional Neural Networks (2018)

[3]. Subashini P., Krishnaveni A.: Image Deblurring Using Back Propagation Neural Network,(2011)

[4]. Neetin K., Shrivastava M., Rahim N,: Image Deblurring Using a Neural Network Approach (2012)

[5]. Gilbert A., Messingher S., Patel A.: Non-Blind Image Deblurring using Neural Networks

[6]. Chakrabarti, A.: A neural approach to blind motion deblurring. In: ECCV (2016)

[7]. Deblurring Images: Matrices, Spectra, and Filtering. Society for Industrial & Applied Mathematics, U.S., 2007.

8. Estimated Budget

The following is the estimated budget for our project:

S.N.	Particulars	Amount (NPR)
1.	Cloud computing with Colab TPU (72 hrs)	37,000/-
2.	Matlab software	5,000/-
3.	Jetbrains IDE platform	28,635/-
4.	Miscellaneous	10,000/-
Grand Total		**80,635/-**