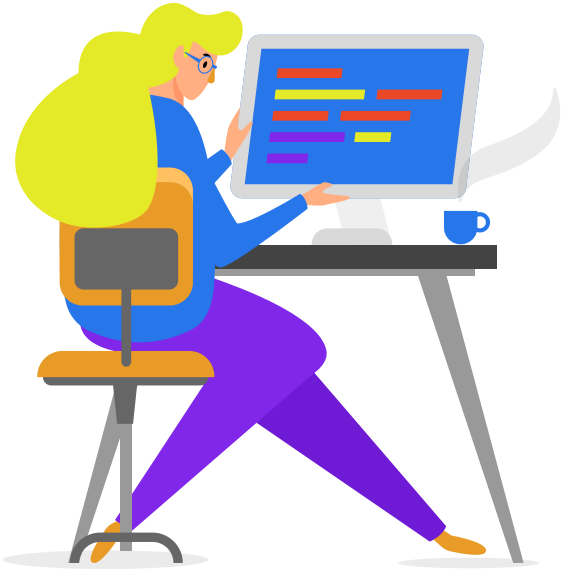
Mahavir Education Trust's
SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE
Chembur, Mumbai - 400 088
UG Program in Computer Engineering

KOFAX

Blur Image Detection

Moiz Rajkotwala - TE3 43
Satyam Shah - TE3 49
Aryan Nandu - TE3 33
Yatish Pant - TE3 35
Shreeshail Devmane - TE3 13
Guide - Prof Shahzia Sayyad

Table of Contents



01 Introduction

02 Problem Statement

03 Objectives

04 Literature Survey

05 Methodology

06 Result

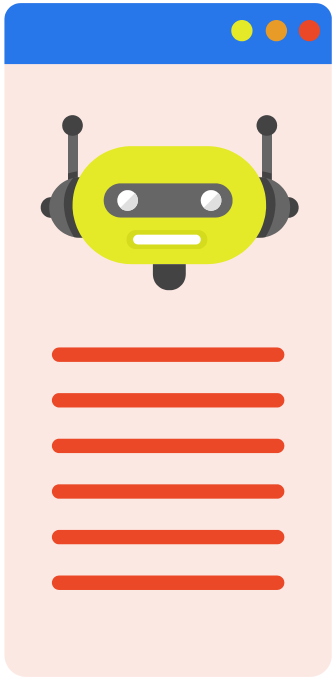
07 Future Scope

08 Conclusion

09 References

Introduction

- Kofax Inc. is an Irvine, California-based intelligent automation software provider.
- Founded in 1985, the company's software allows businesses to automate and
improve business workflows by simplifying the handling of data and documents.
- Kofax develops intelligent automation software for businesses, for applications ranging from print management, to process automation and document/pdf management.



Problem Statement

- Images captured from mobile are sometimes blur and sometimes there are shadow.
- Image quality if not good, server needs to give an error and inform user about the same.
- A server side program that will detect if image is blur.
- Get accuracy of more than 85%

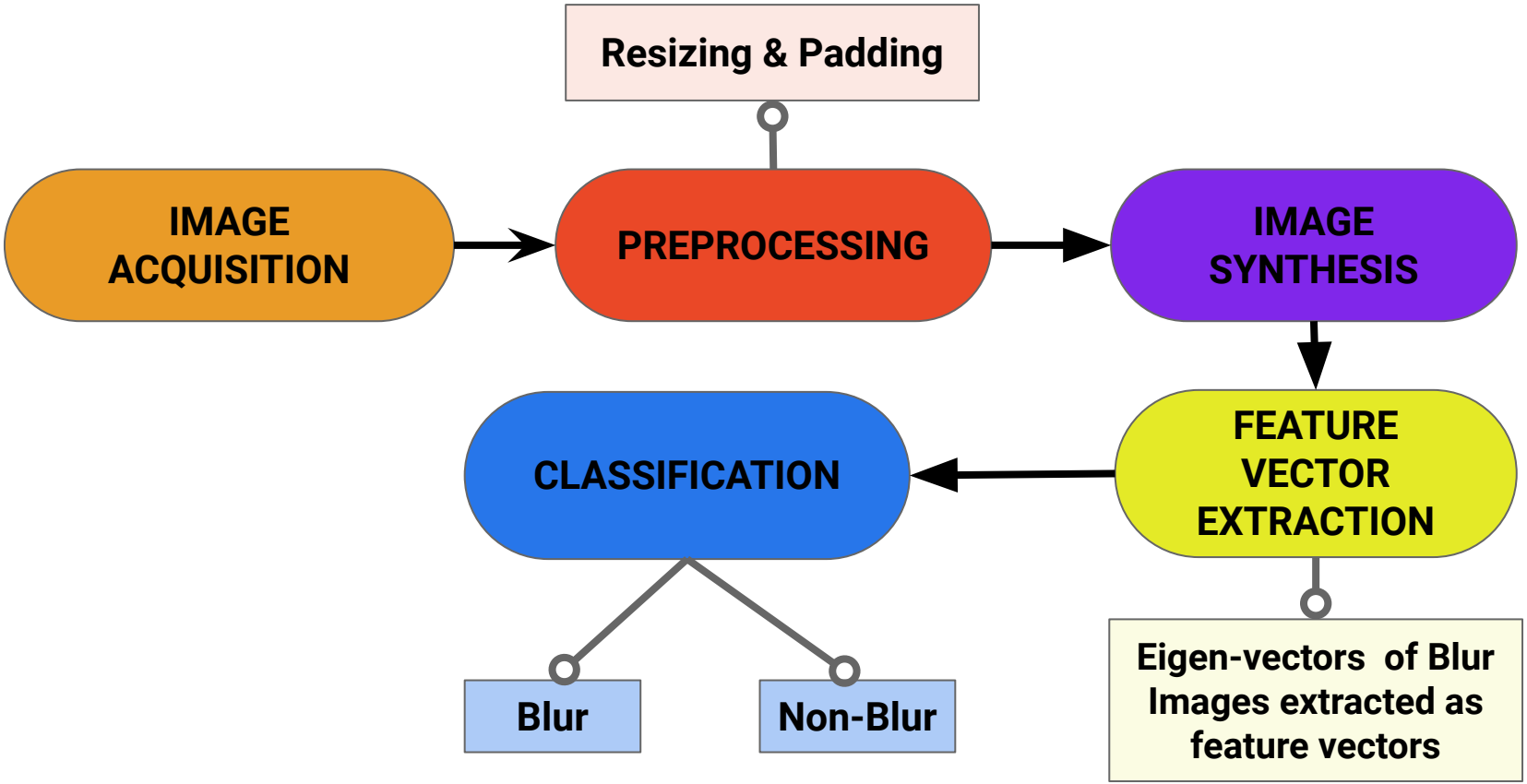
Literature Survey

Sr. No.	Algorithm	Description	Accuracy
1	**Laplacian approach for blur detection [2]**	The Laplace filter is mainly used to define the edge lines in a picture. What is meant here by the edge are the sharp color separations that usually separate objects from the background. The Laplace filter, also known as the Sharpening Filter, uses a window while operating.	80%
2	**Fast Fourier Transform (FFT) [3]**	FFT is applied to convert an image from the spatial domain to the frequency domain. Applying filters to images in frequency domain is computationally faster than to do the same in the spatial domain. FFT takes only (N/2)(log2N) complex multiplications and Nlog2N complex additions.	88.62%

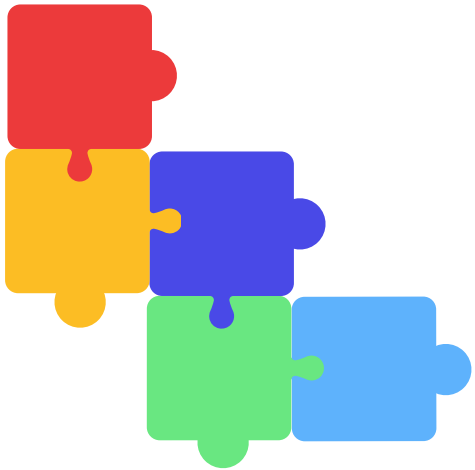
Sr. No.	Algorithm	Workflow	Accuracy
3	**Haar wavelet [4]**	An algorithm is presented for image blur detection with the use of Two Dimensional Haar Wavelet transform (2D HWT). The algorithm classifies an image as blurred or sharp by splitting the image into N x N tiles, applying several iterations of the 2D HWT to each tile, and grouping horizontally, vertically, and diagonally connected tiles with pronounced changes into tile clusters. Images with large tile clusters are classified as sharp. Images with small tile clusters are classified blurred.	15% for blur images 64% for non blur images

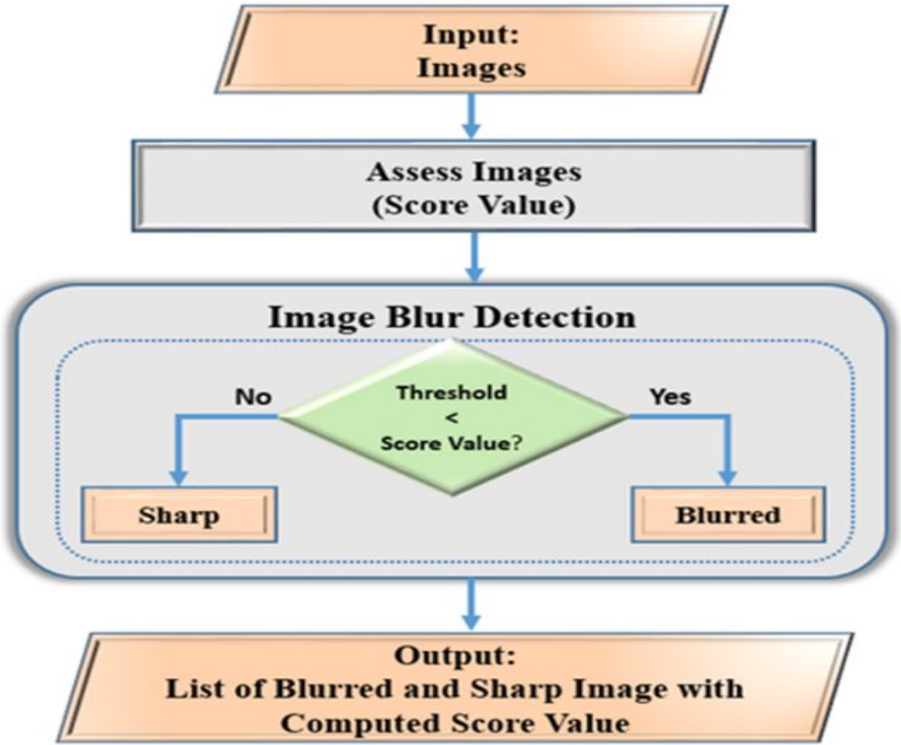
Sr. No.	Algorithm	Description	Accuracy
4	**CNN [6]**	Convolutional neural networks (CNNs or ConvNets) are a specialized form of deep neural networks for analyzing input data that contain some form of spatial structure (Goodfellow et al. 2016) CNNs work by first deriving low-level representations, local edges, and points, and then composing higher level representations, overall shapes, and contours. The name of these deep neural networks is due to the fact that they apply convolutions, a type of linear mathematical operation. CNNs are a powerful tool for image analysis using many layers of filters	We got the highest Accuracy with no pooling

Proposed System



1. Laplacian

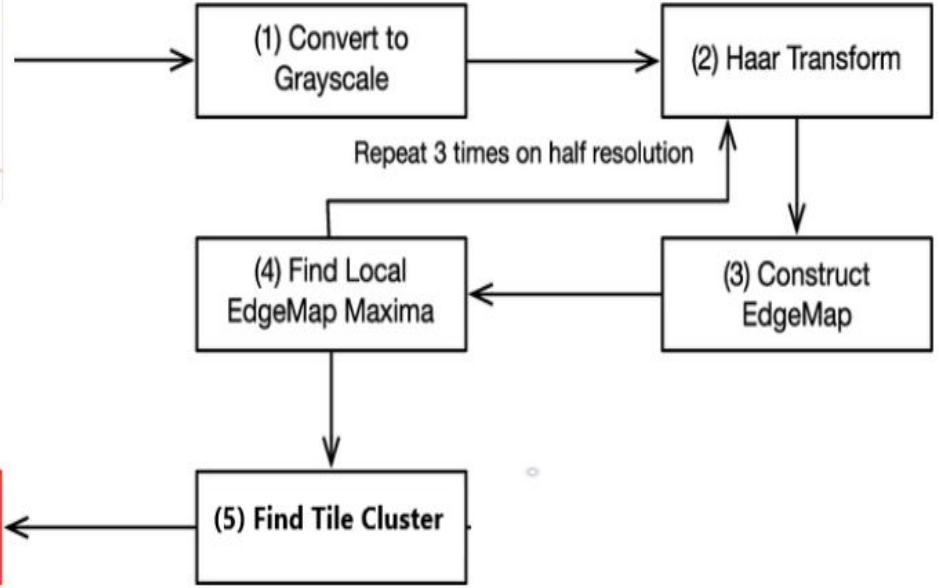




Source : https://www.scitepress.org/Papers/2020/103077/103077.pdf

2. Haar wavelet

Input image



3. FFT (Fast Fourier transform)

Input image

(1) Conver to Grayscale

(2) Grab the dimension and derive the center

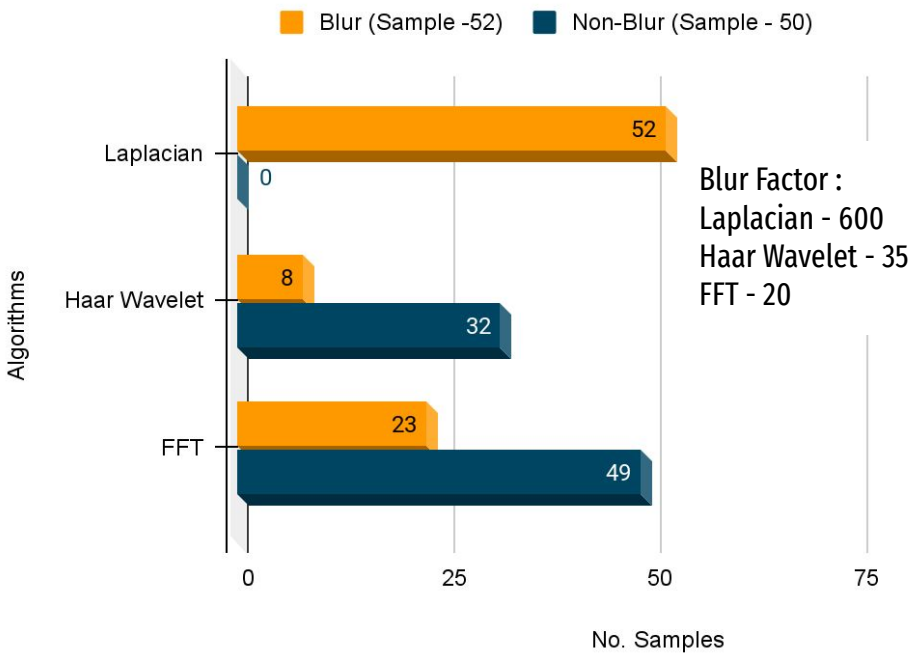
(3) Compute FFT

(4) Shift the component to center

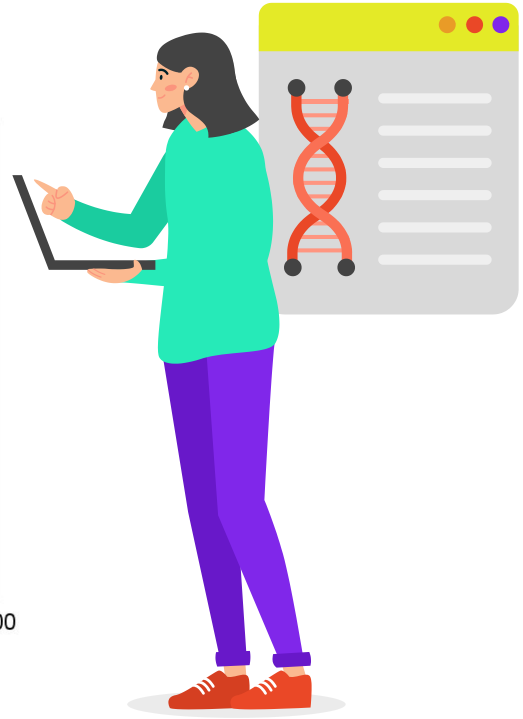
(5) Compute the mean value

(6) Detect Blur

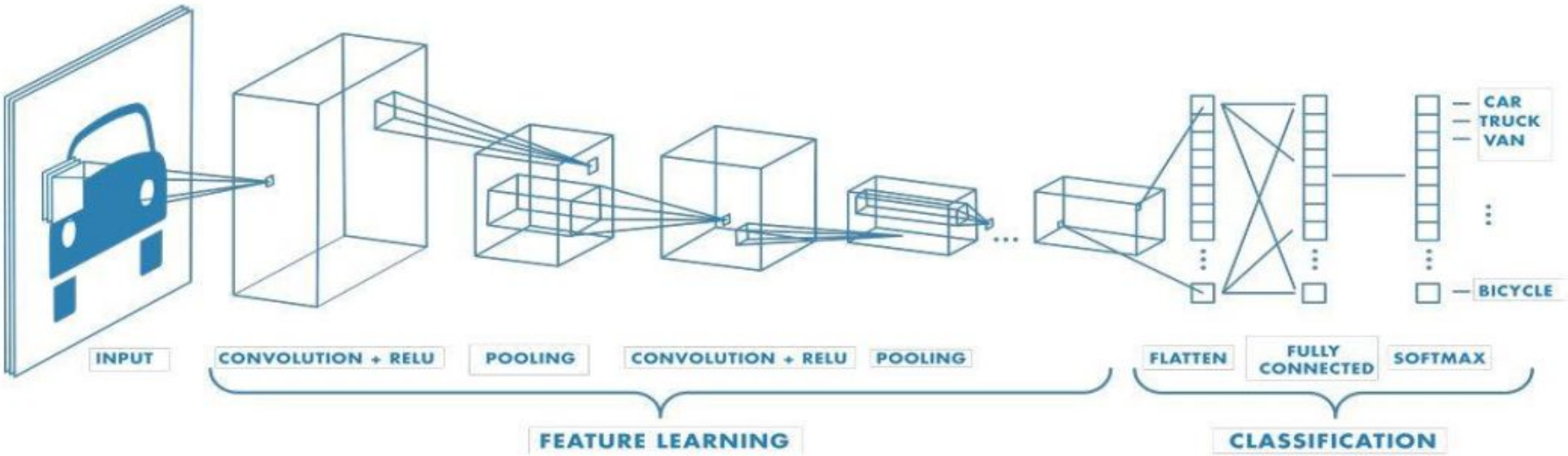
Comparative Result



Blur Factor :
Laplacian - 600
Haar Wavelet - 35
FFT - 20

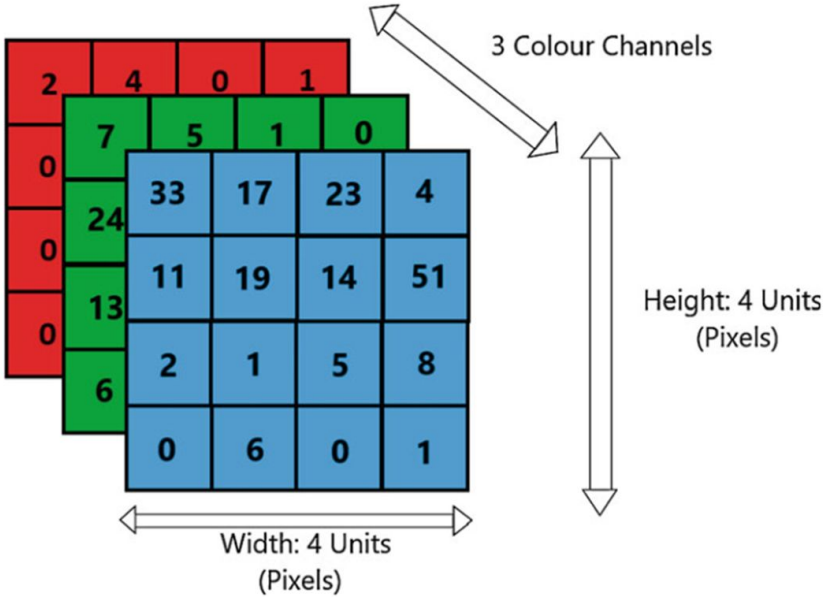


4. CNN (Convolutional Neural Network)

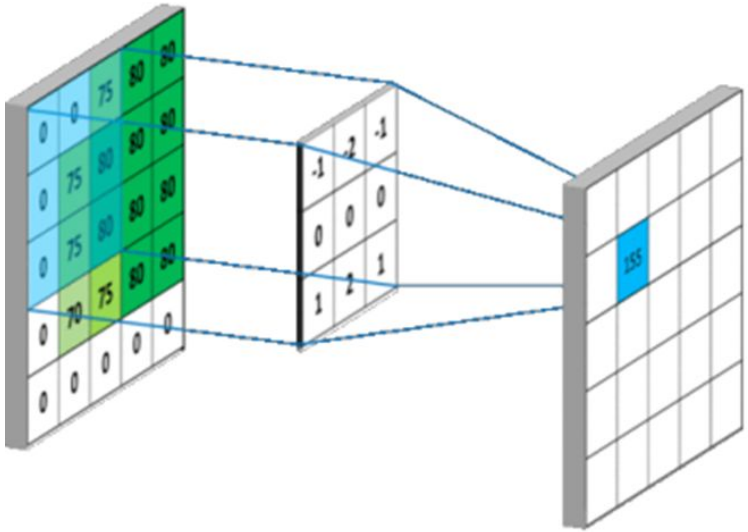


Source : https://www.researchgate.net/publication/348806368

Tensors , Kernels and Convolutional

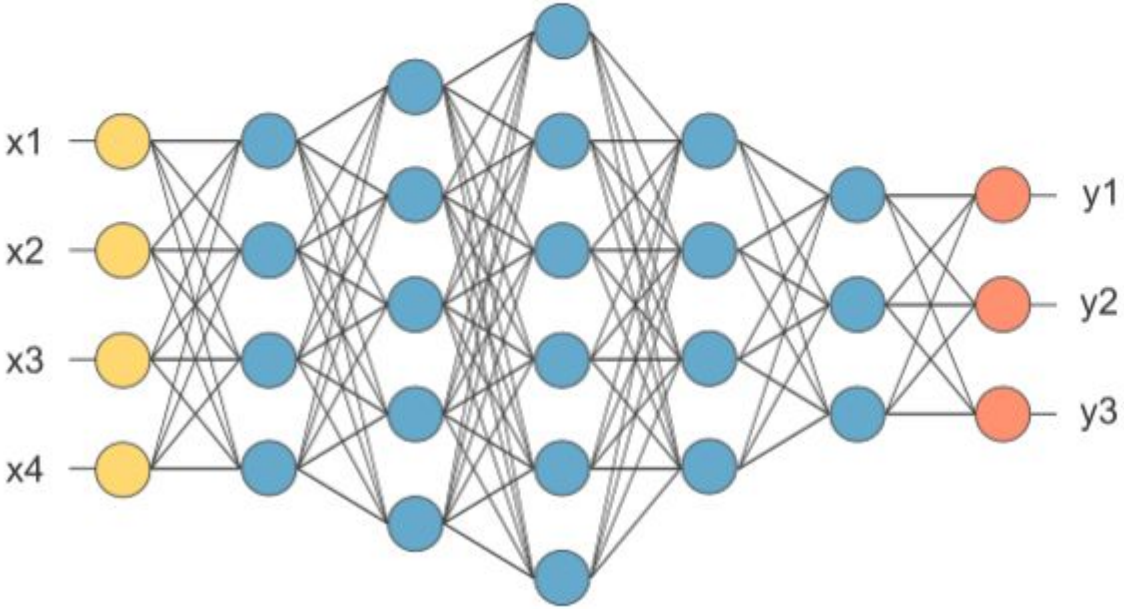


A 3D tensor of RGB Layers of Color Image



Fully Connected Layer

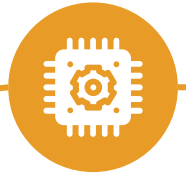
The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like neural network.



Pooling

Pooling simplifies the output by performing nonlinear downsampling, reducing the number of parameters that the network needs to learn

No Pooling

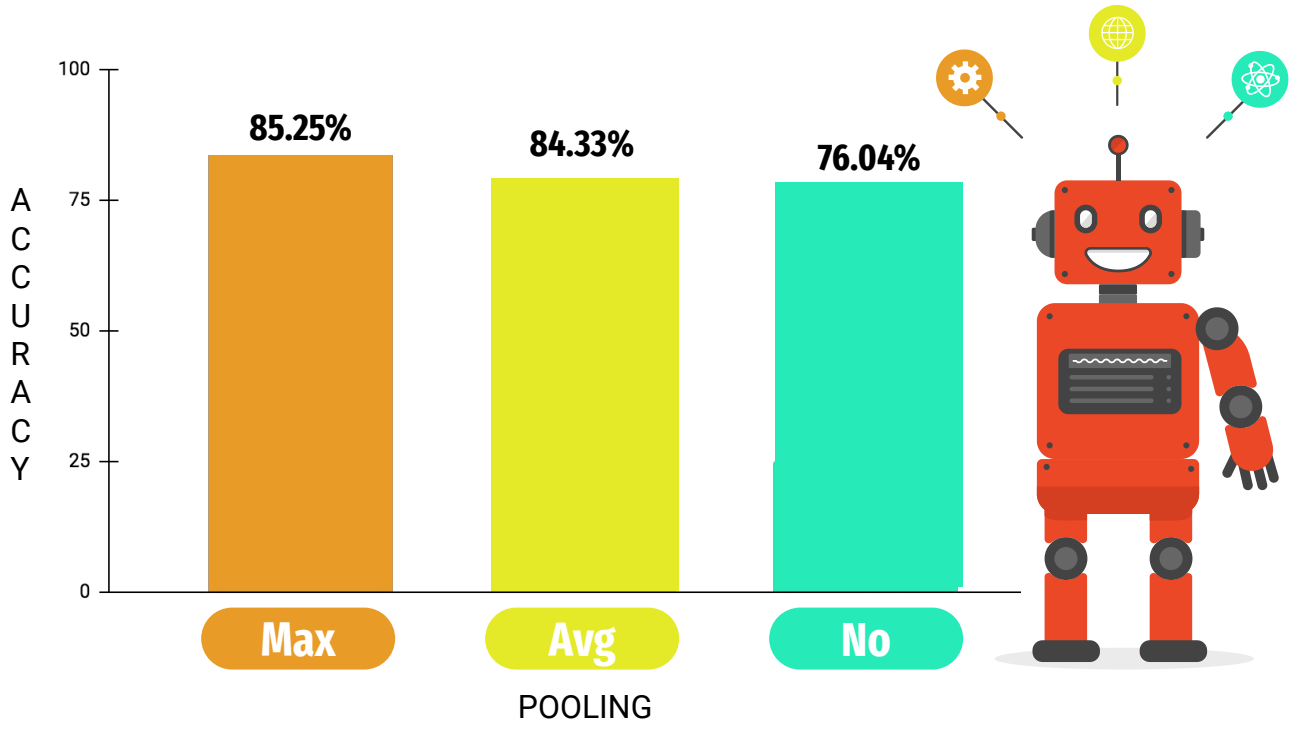


Average Pooling :
Average pooling mostly performs dimensional reduction.

Max pooling:
Max pooling layer returns the maximum values of rectangular regions of inputs, specified by the first argument,poolSize

CNN RESULTS

- Batch size - 16
- Epoch - 25



Analysis of Algorithms

01	Fast Fourier Transform (FFT)	It is necessary to convert an image from the spatial domain to the frequency domain, thus consumes more time in overall process.
02	Laplacian approach	Accuracy for Laplacian Approach for Blur image detection is very less and results are unpredictable.
03	Haar wavelet	Haar cascades tend to be prone to false-positive detections and are not much accurate.
04	CNN	CNN is found to be more accurate and less time consuming with respect to Blur detection than above other algorithms.

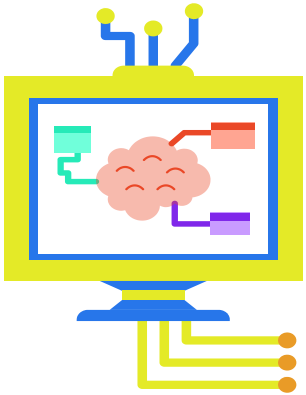
Tensorflow

- ★ Powerful Open-source platform for developing and training machine learning models.
- ★ User friendly API for building and training CNN's
- ★ High-level library for defining and training neural networks
- ★ Also Provides highly optimized run time environment to work with GPUs, making it possible to train very large CNN models in a reasonable time.

Tensorflow Lite

- ★ Designed for mobile devices and embedded systems, perfect for deploying deep learning algorithms on the edge.
- ★ Lightweight and Fast
- ★ Has low latency which is important for real-time applications
- ★ Uses a hybrid approach that enables efficient memory management

TensorFlow Lite



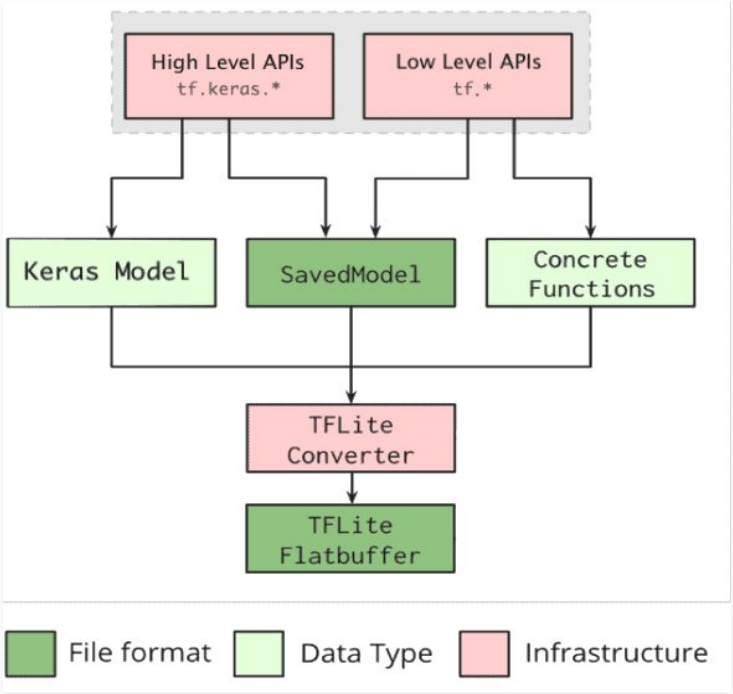
Ways to generate TensorFlow Lite models

01

Creating a TensorFlow Lite model from scratch

Converting a TensorFlow model into a TensorFlow Lite model

02



Model Optimizations

01

Dynamic Range Quantization

Weights are converted to 8-bit precision values. Dynamic range quantization achieves a 4x reduction in the model size in exchange for minimal impacts to latency and accuracy

Models optimized while application development as we get:

- ➔ **Size Reduction**
- ➔ **Latency reduction**
- ➔ **Accelerator Compatibility**

02

In Float-16 quantization

Weights are converted to 16-bit floating-point values. This results in a 2x reduction in model size in exchange for minimal impacts to latency and accuracy

Integer quantization

Optimization strategy that converts 32-bit floating-point numbers (such as weights and activation outputs) to the nearest 8-bit fixed-point numbers

03

Conclusion

- Successfully classified the input image into blur or a non-blur image using CNN method.
- The output layer had 2 neurons which represented the probability of the image being either blur or non-blur.
- CNN is using a training dataset of images which had been manually labelled as blur or non-blur.
- The neural network was able to achieve an accuracy of 99.95% on the training set and 92.25% on the test set.

References

[1] Pagaduan, R. A., Aragon, M. C. R., & Medina, R. P. (2021). iblurdetect: Image blur detection techniques assessment and evaluation study. In Proceedings of the International Conference on Culture Heritage, Education, Sustainable Tourism, and Innovation Technologies-CESIT (pp. 286-291).

[2] Bansal, R., Raj, G., & Choudhury, T. (2016, November). Blur image detection using Laplacian operator and Open-CV. In 2016 International Conference System Modeling & Advancement in Research Trends (SMART) (pp. 63-67). IEEE.

[3] De, K., & Masilamani, V. (2013). Image sharpness measure for blurred images in frequency domain. Procedia Engineering, 64, 149-158.

[4] Tong, H., Li, M., Zhang, H., & Zhang, C. (2004, June). Blur detection for digital images using wavelet transform. In 2004 IEEE international conference on multimedia and expo (ICME)(IEEE Cat. No. 04TH8763) (Vol. 1, pp. 17-20). IEEE.

[6] Scherer, D., Müller, A., & Behnke, S. (2010, September). Evaluation of pooling operations in convolutional architectures for object recognition. In International conference on artificial neural networks (pp. 92-101). Springer, Berlin, Heidelberg.