RV College of Engineering®
Autonomous institution affiliated to Visvesvaraya Technological University, Belagavi
Approved by AICTE, New Delhi, Accredited by NAAC, Bengaluru.
Go, change the world

Improvised visual summarizer using denoising algorithm

A Project Report

Submitted by,

Sahana K S **1RV17EC190**
Rakshata Karlingannavar **1RV17EC119**
Roshni Sen **1RV17EC128**

Under the guidance of

Mr. Subrahmanya K N
Assistant Professor
Dept. of ECE
RV College of Engineering

In partial fulfillment of the requirements for the degree of
Bachelor of Engineering in
Electronics and Communication Engineering
2020-2021

RV College of Engineering®, Bengaluru

(*Autonomous institution affiliated to VTU, Belagavi*)

Department of Electronics and Communication Engineering

.

CERTIFICATE

Certified that the minor project work titled ***Improvised visual summarizer using denoising algorithm*** is carried out by **Sahana K S** (**1RV17EC190**), **Rakshata Karlingannavar** (**1RV17EC119**) and **Roshni Sen** (**1RV17EC128**) who are bonafide students of RV College of Engineering, Bengaluru, in partial fulfillment of the requirements for the degree of **Bachelor of Engineering** in **Electronics and Communication Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2020-2021. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the minor project report deposited in the departmental library. The minor project report has been approved as it satisfies the academic requirements in respect of minor project work prescribed by the institution for the said degree.

Signature of Guide	Signature of Head of the Department	Signature of Principal
Mr. Subrahmanya K N	Dr. K S Geetha	Dr. K. N. Subramanya

External Viva

Name of Examiners	Signature with Date
1.	
2.	

DECLARATION

We, **Sahana K S** , **Rakshata Karlingannavar** and **Roshni Sen** students of seventh semester B.E., Department of Electronics and Communication Engineering, RV College of Engineering, Bengaluru, hereby declare that the minor project titled '**Improvised visual summarizer using denoising algorithm**' has been carried out by us and submitted in partial fulfilment for the award of degree of **Bachelor of Engineering** in **Electronics and Communication Engineering** during the year 2020-2021.

Further we declare that the content of the dissertation has not been submitted previously by anybody for the award of any degree or diploma to any other university.

We also declare that any Intellectual Property Rights generated out of this project carried out at RVCE will be the property of RV College of Engineering, Bengaluru and we will be one of the authors of the same.

Place: Bengaluru

Date:

Name Signature

1. Sahana K S(1RV17EC190)
2. Rakshata Karlingannavar(1RV17EC119)
3. Roshni Sen(1RV17EC128)

ACKNOWLEDGEMENT

We are indebted to our guide, **Mr. Subrahmanya K N**, Assistant Professor, RV College of Engineering . for the wholehearted support, suggestions and invaluable advice throughout our project work and also helped in the preparation of this thesis.

We also express our gratitude to our panel members **Dr.Govinda Raju M.**, Assistant Professor and **Dr. Ramavenkateswaran.N.**, Assistant Professor, Department of Electronics and Communication Engineering for their valuable comments and suggestions during the phase evaluations.

Our sincere thanks to **Dr. K S Geetha**, Professor and Head, Department of Electronics and Communication Engineering, RVCE for the support and encouragement.

We express sincere gratitude to our beloved Principal, **Dr. K. N. Subramanya** for the appreciation towards this project work.

We thank all the teaching staff and technical staff of Electronics and Communication Engineering department, RVCE for their help.

Lastly, we take this opportunity to thank our family members and friends who provided all the backup support throughout the project work.

ABSTRACT

Technology is an evolutionary process that has gained traction in business, academia and government in the recent years. Lectures in classrooms have advanced to the extent of using smart-boards and smart-classrooms. However, there exists an absence of any technological advancement regarding jotting down notes during a presentation or seminar. A visual summarizer is a tool that can be used by attendees of any seminar, presentation or lecture to record vital information in the form of concise notes. The main shortcomings were the lack of GPU and limited RAM availability. Due to these limitations, the denoising autoencoder was trained on a smaller dataset, which made the model vulnerable to overfitting. To avoid this, L2 regularization is used which prevents overfitting of the model by penalising the weights that are large.

The methodology of the project is as follows - to develop and train a denoising autoencoder to deblur the input image, to build an object detection model to detect text from the deblurred image and finally, to extract the detected text and summarize it using extractive summarization. By doing so, the goal of the visual summarizer, that is to generate concise notes from any presentation or lecture, will be achieved.

The software tools used include PyCharm, Google Colab and LabelImg. Software libraries used to train machine learning models are Tensorflow, Keras, OpenCV, PyTesseract, Natural Language Tool Kit (NLTK), matplotlib, pandas and numpy to name a few. A very efficient reconstruction of blurred images and a successful implementation of text extraction and summarization on them was achieved. The accuracy of the trained models can be further improved by increasing the size of the image dataset and training for more number of epochs. The future scope of this project could be to generate personalized summaries specific to each user - the extent of details in the summary could be customized by the user depending on their expertise in the topic of discussion.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CE Cross Entropy Loss

CNN Convolutional Neural Networks

DAE Denoising Autoencoder

NLP Natural Language Processing

NLTK Natural Language Tool Kit

OpenCV Open source computer vision

R–CNN Region based Convolutional Neural Networks

Chapter 1

Introduction to Improvised Visual Summarizer using Denoising Algorithm

CHAPTER 1

INTRODUCTION TO IMPROVISED VISUAL SUMMARIZER USING DENOISING ALGORITHM

1.1 Introduction

As of today, technology is ubiquitous and has advanced at an unprecedented rate in several sectors with more than half the world's population living hand-in-hand with technology. From smart watches to smart phones to smart cities, the embodiment of artificial intelligence in all the day-to-day activities no longer looks like a distant dream. Lectures in classrooms have also advanced to the extent of using smart-boards and smart-classrooms; the developments in jotting down notes in such scenarios has, however, not advanced at the same pace.

Along with reading and listening, taking notes can tend to become a passive activity where one doesn't quite register what the speaker is talking about but instead aimlessly writes down as much as they can keep up with. If one's approach to note taking involves trying to write down, word-for-word, just about everything that the teacher says, they will be more involved in getting words on paper than in focusing their attention and asking questions about the points that are important. An effective approach for successful classroom learning is to be an active listener as well as take well-organized and brief, yet explicit, notes; making them complete enough to provide an overview of the entire lecture.

In the problem statement being tackled, the target audience includes the attendees of any seminar, presentation or lecture, be it students or the public in general, attending important conferences and talks. More often than not, providing complete undivided attention to the speaker proves to be difficult at these seminars as the attendee may be preoccupied with the objective of jotting down pointers and making notes for future reference. It is during this process that several essentials in the speaker's delivery are missed out. Keeping this in mind, this project delves into the implementation of an efficient visual summarizer that provides a solution to this problem. Be it visual images of PowerPoint slides or handwritten material that is presented in the seminars, this tool

will provide a smart solution of summarizing the entire presentation and generating an output of the same.

1.2 Problem statement

The process of taking notes during a presentation or seminar tends to become a passive activity while one tries to listen and actively participate in discussions regarding the topic in question. Hence, there is a requirement to aid attendees of lectures and presentations in the process of recording vital information, in a concise format, dispensed during the addressal.

1.3 Objectives

The objectives of the project are

1. To develop and train a denoising autoencoder to deblur the input image.
2. To build an object detection model to detect text from the deblurred image.
3. To extract the detected text and summarize it using extractive summarization.

1.4 Literature Review

Literature review, an integral part of every project, provides insight into how the proposed research is related prior research and demonstrates the originality and relevance of one's research problem as well as justifies the proposed methodology. Autoencoder is very popular neural network for image processing. Denoising autoencoder is an important autoencoder because in some tasks a preprocessed image is needed to get less noisy result. The research in [1] describes ways to analyze noisy images and how to reduce that noise. The best result achieved after training is for 186 samples and this process has taken 2185.99 seconds and 95% of 2048 frames have been denoised in 84.26 seconds. Furthermore, it was established that this method is 3.3 times faster than conventional methods.

[2] discusses the powerful learning ability of deep learning based object detection and its potential to handle occlusion, clutter and low-resolution. Further, it also discusses Convolutional Neural Networks and its advantages over traditional methods – hidden factors of input data can be extracted, exponentially augmented expressive capability, optimization of relative tasks. It also provides a brief review of salient object detection,

face detection and pedestrian detection. Finally it demonstrates a few promising future directions and tasks such as multi-task joint optimization, scale adaptation, cascade networks and 3D object detection.

[3] primarily deals with the trends and vast range of research areas of deep-learning based object detection. Some of these trends are as combining one stage (fast but lower accuracy) and two stage detectors (higher accuracy but time consuming), video object detection include obstacles like motion blur, motion target ambiguity, smart targets, etc., versatility of multi-domain object detection. This paper also discusses unsupervised object detection models for intelligent detection mission and advanced medical biometrics to analyze retinal images and speech patterns.

[4] is the base paper for the project. The objective of this paper was to enable the users to give their undivided attention to the lecture taking place, without them fretting over the key details, as the audio-visual summarizer will permit the users to always go back and refer to the lecture. The proposed solution included 4 vital steps - integration of Raspberry Pi with camera and microphone, implementing a text detection and extraction model, speech to text conversion using LAS model and summarizing the extracted text from images and audio using Natural Language Toolkit (NLTK).

The traditional optical character recognition technology (OCR) requires the text neat layout and neatness and background clean, and industrial production often fail to meet such standards. [5] proposes a new method based on convolution neural network (Faster RCNN) that aims to improve the correctness of text recognition. When compared with the conventional detection method, the correct rate of recognition based on Faster RCNN model can reach 90.4%, and the correctness rate is 88.9%. When compared with the traditional OCR, the method proposed in this paper is relatively stable, and the accuracy is relatively high.

[6] provides a detailed analysis of Optical Character Recognition technology. The objective of OCR is to achieve modification or conversion of any form of text or text-containing documents such as handwritten text, printed or scanned text images, into an editable digital format for deeper and further processing. The challenges faced by OCR are scene complexity, conditions of uneven lighting, skewness (rotation), blurring and degradation, aspect ratios, fonts, multilingual environments. There are various phases of OCR such as preprocessing, segmentation, normalization, feature extraction, classifica-

tion and post processing. Finally, this paper discusses the applications of OCR, namely handwriting recognition, receipt imaging, legal industry, banking, healthcare, automatic number plate recognition, etc.

The objective of [7] is a summarization system that produces a summary for a given web document based on sentence importance measures. It is an efficient approach for single document summarization which uses the two sentence importance measures. The first is the frequency of the terms in the sentence and its similarity to the other sentences. The second is sentences are ranked according to their respective scores and the sentences with top ranks are selected for summary. The summary is evaluated by using 'recall' evaluation measure.

1.5 Brief Methodology of the project

The first step is to train a denoising autoencoder for deblurring the input images. The following move is to prepare an image dataset for training the object detection model, Region based Convolutional Neural Networks (Region Based Convolutional Neural Networks). Running the image through a Convoluted Neural Network (CNN) will generate a Feature Map. On running the Activation Map through a separate network, called the Region Proposal Network (RPN), interesting boxes/regions will be generated as outputs. The predicted region proposals are then reshaped using a Region of Interest Pooling (RoIP) layer following which it is passed through R–CNN. R–CNN has two functions:

1. Classify proposals into one of the classes
2. Better adjust the bounding box for the proposal according to the predicted class i.e. group proposals of similar class to get final bounding box.

Passing the blurred images through a denoising autoencoder before passing through the object detection model will help achieve accurate detection of the text from the image which will then extracted using Tesseract OpenCV module and summarized using Natural Language Toolkit.

1.6 Assumptions made / Constraints of the project

The assumptions made for the execution of the project are as follows:

1. **Hardware Constraint:** Accuracy of the project is impacted due to the lack of GPU and limited RAM.

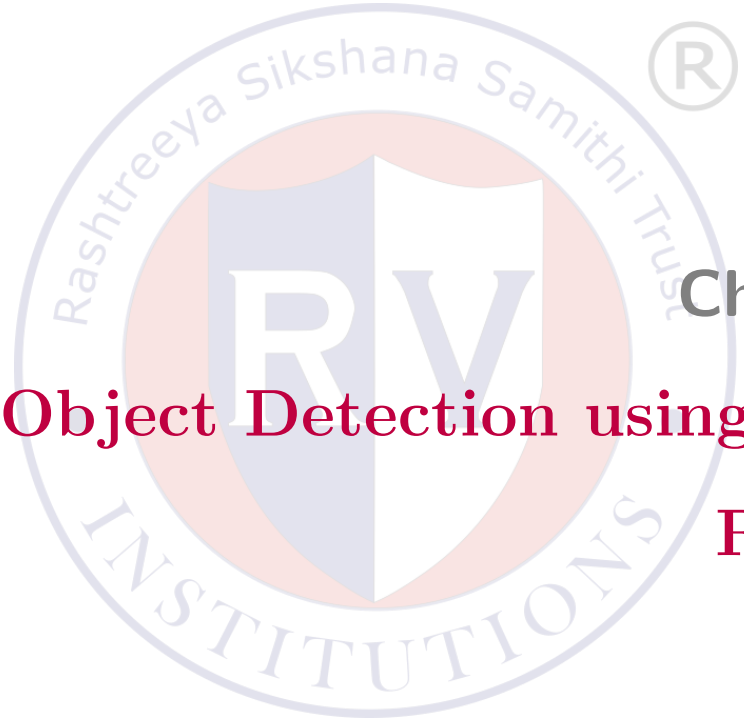
2. **Limited dataset:** The data set prepared is self-made and a limited number of images were collected. With an increased number of images in the dataset will lead to better accuracy.

3. **Limitation with respect to blur:** The Denoising Autoencoder is trained for Gaussian blur. For normal blurring, such as that taken from a camera, the output of the autoencoder will not be inaccurate.

1.7 Organization of the report

This report is organized as follows. Write the discussions in each chapter. A sample is as follows.

- Chapter 2 discusses the fundamentals of Convolutional Neural Networks.
- Chapter 3 introduces object detection and discusses the Faster R–CNN model used as well as the model summary.
- Chapter 4 introduces the Denoising Autoencoder and gives a description of the model summary and training details.
- Chapter 5 gives a description of text extraction and summarization methods.
- Chapter 6 discusses the results obtained after each stage and its analysis.
- Chapter 7 concludes the report.



Chapter 2

Object Detection using Faster R–CNN

CHAPTER 2

OBJECT DETECTION USING FASTER R–CNN

In Convolutional Neural Networks, the goal is to provide an image as an input and generate an output that determines the probability of the image belonging to a certain class. This chapter discusses the fundamentals of CNN as well as its various layers, i.e the Convolutional Layer, the Pooling Layer and the Fully Connected Layer. Object detection is a computer vision task that involves predicting where the required objects are in an image. This chapter gives an introduction to object detection. It further gives details on one of the most efficient object detection algorithms, the Faster R–CNN model and then gives the summary of the trained object detection model.

2.1 Introduction to Convolutional Neural Networks

Artificial Intelligence has contributed in a monumental manner to bridge the gap between humans and computer abilities. To make great things possible, researchers work on various facets of the area, the domain of Computer Vision being one of several such fields. The goal for this field is to allow machines to view the world as humans do, interpret it in a similar way and even use information for a variety of tasks, such as recognition of images and videos, reconstruction of media, recommendation systems, processing of natural languages, and so on. With time, the advances in Computer Vision with Deep Learning have been developed and refined, predominantly through a specific algorithm, a Convolutional Neural Network.

A Convolutional Neural Network (ConvNet/CNN) is a deep learning algorithm that can take an input image, assign significance to various aspects/objects in the image and be able to distinguish one from the other. In comparison to other classification algorithms, the pre–processing required in a CNN is much lower. In CNN, filters are not hand-engineered – with enough training, they have the ability to learn these filters.

The architecture of Convolutional Neural Networks differs from that of regular Neural Networks. In case of regular Neural Networks, an is transformed by passing it through multiple hidden layers where each layer consists of a set of neurons and is entirely connected to all neurons in the layer before, following which there is a final fully-connected layer — the output layer — that represents the predictions. There is a slight difference in case of Convolutional Neural Networks. Here, the layers are organised in 3 dimensions:

width, height and depth. Further, the neurons in one layer do not connect to all the neurons in the next layer but only to a small region of it. Ultimately, the final output will be reduced to a single vector of probability scores, organized along the depth dimension.

2.2 Working of Convolutional Neural Networks

A CNN has several layers for processing an image which are discussed below.

2.2.1 Convolution Layer

Convolution layer is the first layer to extract features from an input image. An image is nothing but a matrix of pixel values. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel to produce a feature map. Convolution is executed by sliding the filter over the input. At every location, a matrix multiplication is performed between the image matrix and the filter matrix and the result is summed onto the feature map as shown in figures 2.1 and 2.2.

Figure 2.1: Image matrix multiplied with kernel or filter matrix

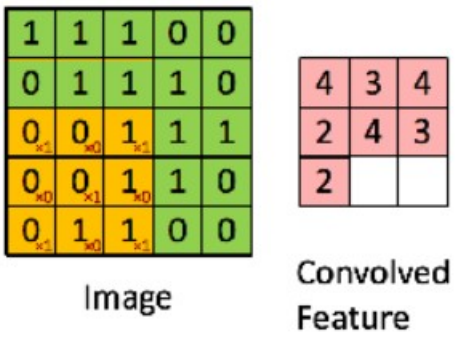


Figure 2.2: Output matrix

Operations such as edge detection, blur and sharpen can be achieved by convolution of an image with different filters.

2.2.2 Pooling Layer

When the images are too large, pooling layers can reduce the number of parameters. Spatial pooling, also called subsampling or downsampling, reduces the dimensionality of each map but retains important information. Spatial pooling can be of several types:

1. Max Pooling
2. Average Pooling
3. Sum Pooling

Max Pooling returns the maximum value from the portion of the image covered by the kernel. Average Pooling returns the average of all the values from the portion of the image covered by the Kernel. Sum of all elements in the feature map is called Sum Pooling. Figure 2.3 shows the operation of max pool with $2X2$ filters and stride 2.

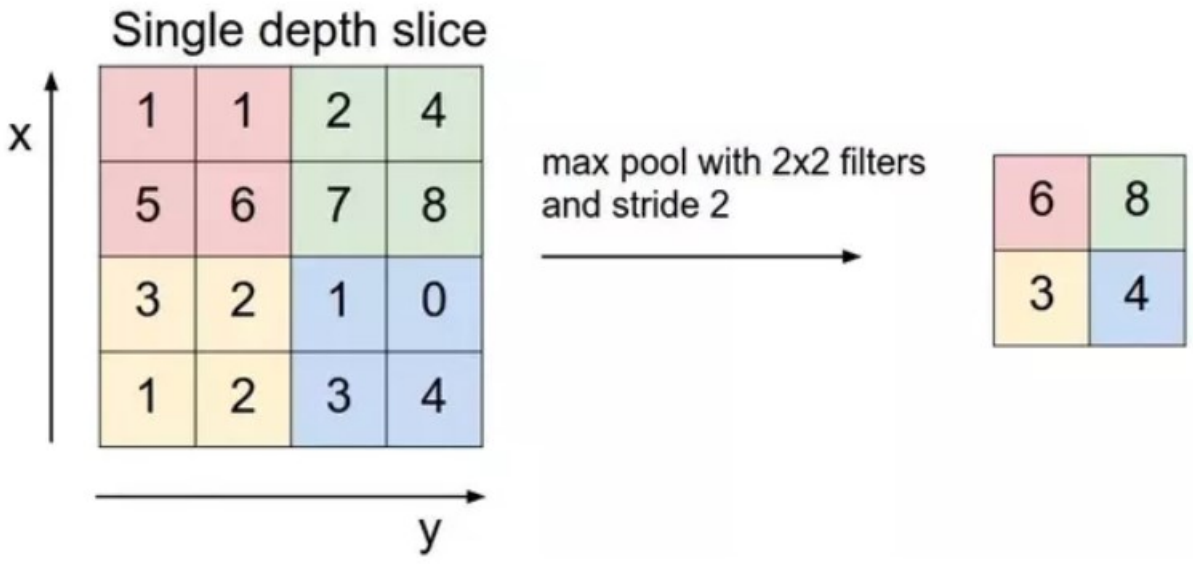


Figure 2.3: Max Pooling

2.2.3 Fully Connected Layer

The input to the fully connected layer is the output from the final Pooling or Convolutional Layer, which is flattened (unroll the output of final (and any) Pooling and

Convolutional Layer, which is a 3-dimensional matrix, values into a vector) and then fed into the fully connected layer.

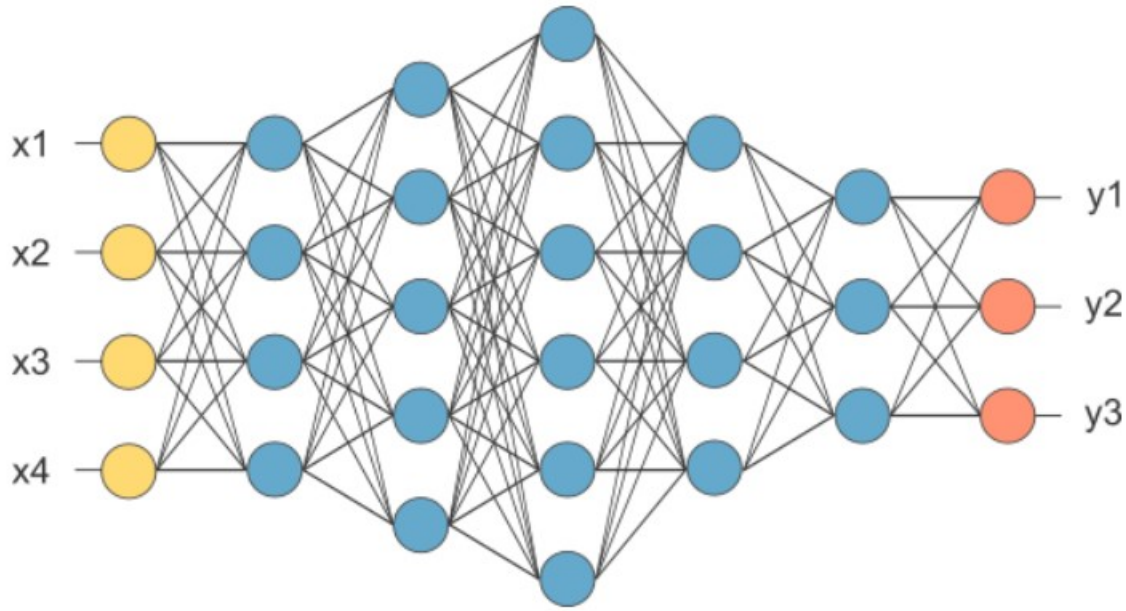


Figure 2.4: After pooling layer, flattened as FC layer

As shown in figure 2.4, the feature map matrix will be converted as vector (x1, x2, x3, ...). With the Fully Connected Layers, these features are combined together to create a model. Finally, there is an activation function such as softmax or sigmoid to classify the outputs as car, truck, cat, dog, etc.

2.3 Other parameters associated with CNN layers

2.3.1 Stride

It is the number of pixels shifts over the input matrix. The filter is moved 1 pixel at a time when the stride is 1. The filter is moved 2 pixels at a time when the stride is 2 and so on. Figure 2.5 shows how convolution would work with a stride of 2.

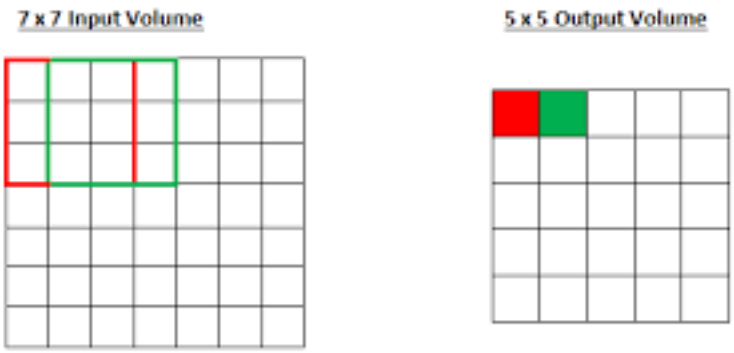


Figure 2.5: Stride

2.3.2 Padding

There are times when the filter does not perfectly fit the input image. In such cases, there are two options:

1. Pad the picture with zeros (zero-padding) so that it fits
2. Drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image.

2.3.3 Non Linear Activation function

Activation functions are mathematical equations that determine the output of a neural network. The function is attached to each neuron in the network, and determines whether it should be activated ("fired") or not, based on whether each neuron's input is relevant for the model's prediction. An example for activation function is ReLU (Rectified Linear Unit) which is shown in figure 2.6. ReLU is a non linear activation function defined as f(x) = max(0,x). It will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks since the problem of having negative weights can be avoided as negative values will be scaled to 0.

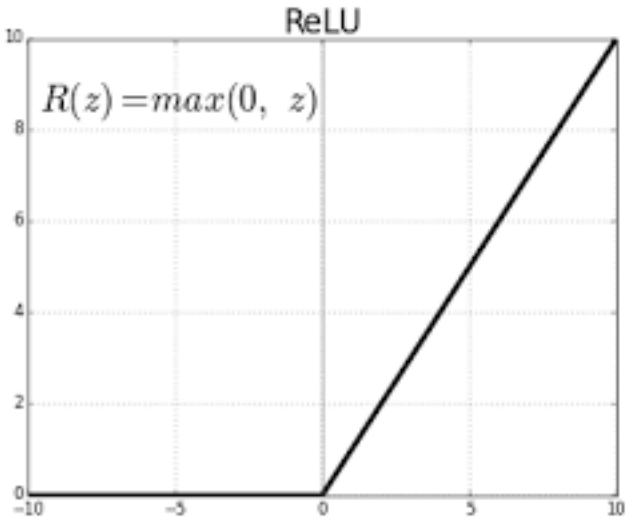


Figure 2.6: ReLU activation Function

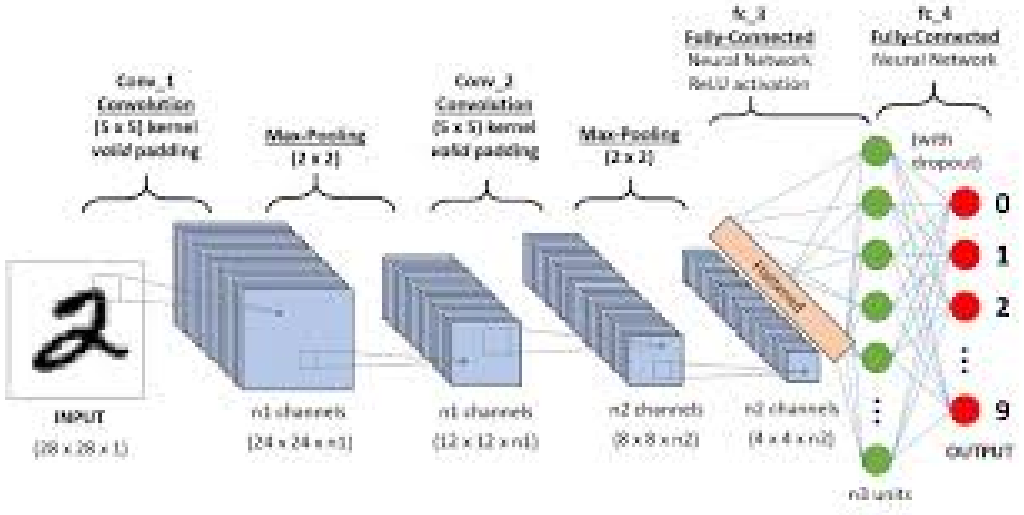


Figure 2.7: CNN architecture

Figure 2.7 shows the entire CNN architecture which can be briefly described as follows. The input image is provided to the convolution layer. Parameters are chosen, filters with strides are applied along with padding, if required. Convolution on the image is performed and a non linear activation is applied to the matrix. Pooling is performed to reduce image dimension. The output of this stage is flattened and fed into a fully connected layer (FC Layer). This layer outputs the class to which the input image belongs.

2.4 Introduction to Object Detection

Image Processing is a technique that plays out a couple of assignments in a picture, to generate a picture which is improved or to extricate information that is supportive

from it. A kind of sign handling it is, in which info is a picture and yield may be picture or highlights related with that image. In the recent days, image preparing is one of the rapidly creating developments. It shapes focus examine district inside structure and disciplines too of programming building. Picture handling incorporates fundamentally the three accompanying stages, bringing in the picture by means of securing picture instruments, controlling and breaking down the picture and yield in which the end data can be changed picture or report that depends on picture examination. There are couple of types of strategies used for picture preparing explicitly, simple and computerized image handling. Image specialists use few nuts and bolts of comprehension while simultaneously using these visual procedures. Advanced Image handling strategies help in charge of the modernized pictures by using PCs. A library fundamentally focused on current-time computer vision of programming capacities is Open source computer vision (OpenCV). OpenCV bolsters a few models from profound learning structures like TensorFlow, Torch, PyTorch (subsequent to changing over to an ONNX model) and Caffe as indicated by a characterized rundown of upheld layers. It advances OpenVisionCapsules, which is a versatile configuration, perfect with every other arrangement.

Computer vision is an interdisciplinary field that has been gaining huge amounts of traction in the recent years(since CNN) and self-driving cars have taken centre stage. Another integral part of computer vision is object detection. Object detection aids in pose estimation, vehicle detection, surveillance etc. The difference between object detection algorithms and classification algorithms is that in detection algorithms, there is an attempt to draw a bounding box around the object of interest to locate it within the image. Also, there might not be necessarily just one bounding box in an object detection case, there could be many bounding boxes representing different objects of interest within the image.

Object detection [2] is a computer vision technique that uses image processing technique which identifies and locates objects in an image or video. With this kind of identification and localization, object detection can be used to count objects in a scene and determine and track their precise locations, all while accurately labeling them.

Object recognition is a general term to describe a collection of related computer vision tasks that involve identifying objects in digital photographs. Image classification involves predicting the class of one object in an image. Object localization refers to identifying

the location of one or more objects in an image and drawing abounding box around their extent. Object detection combines these two tasks and localizes and classifies one or more objects in an image.

As such, computer vision tasks can be distinguished as follows:

1. Image Classification: Predict the type or class of an object in an image.
   (a) Input: An image with a single object, such as a photograph.
   (b) Output: A class label (e.g. one or more integers that are mapped to class labels).
2. Object Localization: Locate the presence of objects in an image and indicate their location with a bounding box.
   (a) Input: An image with one or more objects, such as a photograph.
   (b) Output: One or more bounding boxes (e.g. defined by a point, width, and height).
3. Object Detection: Locate the presence of objects with a bounding box and types or classes of the located objects in an image.
   (a) Input: An image with one or more objects, such as a photograph.
   (b) Output: One or more bounding boxes (e.g. defined by a point, width, and height), and a class label for each bounding box.

The dependencies of these computer vision tasks can be modelled as shown in figure 2.8.

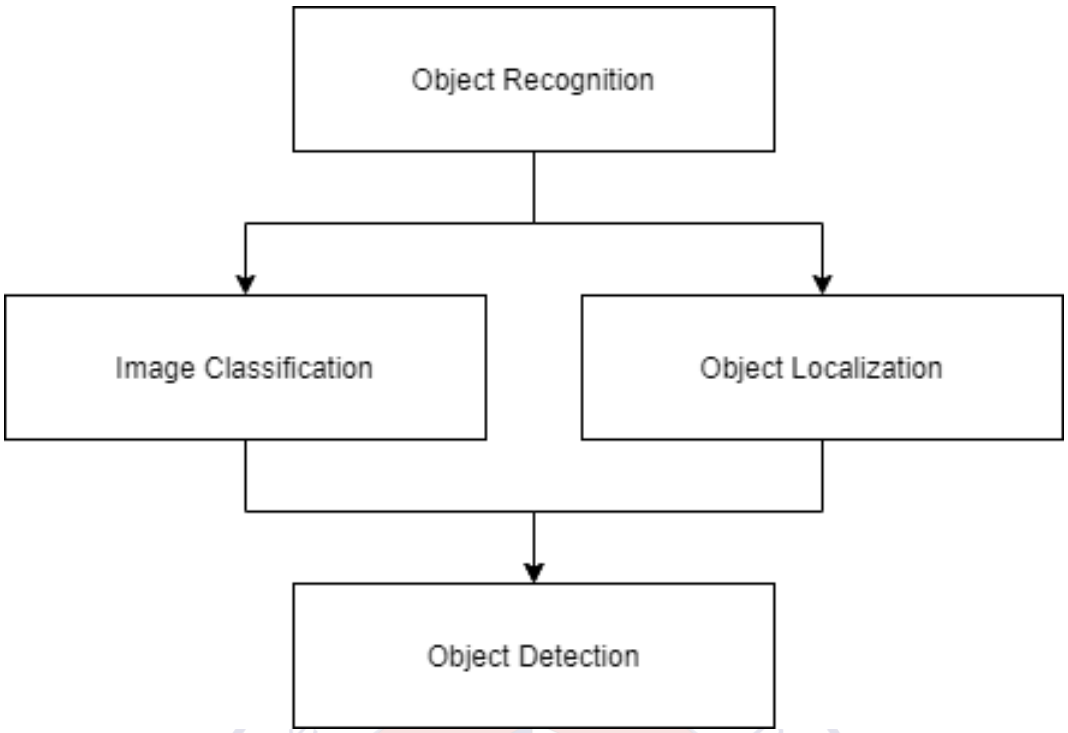


Figure 2.8: Object Detection

A naive approach to solve this problem would be to take different regions of interest from the image, and use a CNN to classify the presence of the object within that region. The problem with this approach is that the objects of interest might have different spatial locations within the image and different aspect ratios. Therefore, algorithms like R–CNN, YOLO etc have been developed to find these occurrences and find them fast.

2.5 Introduction to Region based Convolutional Neural Networks

The problem the R–CNN system tries to solve it is to locate objects in an image (object detection). R–CNN "Region-based Convolutional Neural Networks". The main idea is composed of two steps.

1. First, using selective search, it identifies a manageable number of bounding-box object region candidates ("region of interest" or "RoI").

2. It then extracts CNN features from each region independently for classification.

To make R–CNN faster, the training procedure was improved by unifying three independent models into one jointly trained framework and increasing shared computation

results, named Fast R–CNN. Instead of extracting CNN feature vectors independently for each region proposal, this model aggregates them into one CNN forward pass over the entire image and the region proposals share this feature matrix. Then the same feature matrix is branched out to be used for learning the object classifier and the bounding-box regressor. In conclusion, computation sharing speeds up R–CNN. Fast R–CNN is much faster in both training and testing time. However, the improvement is not dramatic because the region proposals are generated separately by another model and that is very expensive.

An intuitive speedup solution is to integrate the region proposal algorithm into the CNN model. Region based Convolutional Neural Networks [5] is doing exactly this: construct a single, unified model composed of RPN (region proposal network) and fast R-CNN with shared convolutional feature layers.

Region based Convolutional Neural Networks (R–CNN) is composed of 3 neural networks — Feature Network, Region Proposal Network (RPN), Detection Network.

2.6 Software Setup

1. Tensorflow: It is an open source stage from start to finish for AI. An exhaustive, biological system of instruments that is adaptable is included in it, network assets and libraries that lets ML to be pushed best in class by scientist and ML fueled applications are effectively assembled and conveyed by engineers.

2. LabelImg tool: LabelImg is a free, open source tool for graphically labeling images. It's written in Python and uses QT for its graphical interface. It is an an open-source image labeling tool for training classifiers.

2.7 Text Detection model using Region based Convolutional Neural Networks

The object detection was successfully performed using TensorFlow object detection. COCO models provided pre-defined models for object detection. A variety of models with pre-assigned set of initial weights and pre-made architectures allowed flexibility in choosing the models based on training results. The faster_rcnn_v2_coco model [4] offered a considerably greater accuracy. Here the R–CNN [2] model applies high-capacity convolutional neural networks so that a fixed-length feature vector from each region can

be extracted which is then fed to a set of class-specific linear SVMs. The Fast R–CNN and Region based Convolutional Neural Networks [5] have made further evolution on the pipeline of object detection. Following the pioneering R–CNN, Fast/Faster R–CNN uses convolutional layers, initialized with discriminative pretraining for ImageNet classification, to extract region-independent features followed by a region wise multilayer perceptron (MLP) for classification. Datasets were made using LabelImg, an open-source image labeling tool for training classifiers.

The faster_rcnn_inception_v2_coco provides a special inception block to reduce the feature map size. These size reduction blocks have parameters specifically to maintain alignment of the feature map size in the concatenation layer. Figure 2.9 given the structure of the RCNN model used for object detection.

Layer (type)	Output Shape	Param #
Conv2d+ReLU-1	[224,224,3]	23,296
Conv2d+ReLU-2	[224,224,64]	25,625
MaxPool2d-3	[112,112,128]	0
Conv2d+ReLU-4	[112,112,128]	307,392
Conv2d+ReLU-5	[112,112,128]	235,985
MaxPool2d-6	[56,56,256]	0
Conv2d+ReLU-7	[56,56,256]	663,936
Conv2d+ReLU-8	[56,56,256]	775,657
Conv2d+ReLU-9	[56,56,256]	884,992
MaxPool2d-10	[28,28,512]	0
Conv2d+ReLU11	[28,28,512]	590,080
Conv2d+ReLU-12	[28,28,512]	534,298
Conv2d+ReLU-13	[28,28,512]	987,482
MaxPool2d-14	[14,14,512]	0
Conv2d+ReLU-15	[14,14,512]	13,429,632
Conv2d+ReLU-16	[14,14,512]	37,752,832
Conv2d+ReLU-17	[14,14,512]	61,439,529
MaxPool2d-18	[7,7,512]	0
FullyConnected+ReLU-19	[1,1,4096]	16,781,312
FullyConnected+ReLU-20	[1,1,4096]	20,648,311
FullyConnected+ReLU-21	[1,1,1000]	23,837,362
Softmax-22	[1,1,1000]	4,097,000

Total params: 188,014,721
Trainable params: 188,014,721
Non-trainable params: 0

Figure 2.9: Model summary of the text detection model

In this chapter, it was established that Region based Convolutional Neural Networks is faster in both training and testing and is much more efficient than the traditional methods like CNN (Convolutional Neural Networks) and OCR (Optical Character Recognition) for object detection and also the details of the model used to build the text detector.

Chapter 3

Denoising Autoencoder

CHAPTER 3

DENOISING AUTOENCODER

Denoising autoencoders are used to denoise an input image. They are widely used as first pre–processing step in various image processing applications. This chapter gives a brief introduction to denoising autoencoders. It further gives a description of the model summary and training details.

3.1 Autoencoding

"Autoencoding" is a data compression algorithm where the compression and decompression functions are 1) data-specific, 2) lossy, and 3) learned automatically from examples rather than engineered by a human. These compression and decompression functions are usually implemented using neural networks.

1. Autoencoders are data-specific : they will only be able to compress data similar to what they have been trained on. An autoencoder trained on pictures of faces would do a rather poor job of compressing pictures of trees, because the features it would learn would be face-specific. Hence, one model cannot be generalised to all applications. They are application specific.

2. Autoencoders are lossy : the decompressed outputs will be degraded compared to the original inputs (similar to MP3 or JPEG compression). This differs from lossless arithmetic compression.

3. Autoencoders are learned automatically from data examples : it means that it is easy to train specialized instances of the algorithm that will perform well on a specific type of input. It doesn't require any new engineering, just appropriate training data.

The aim of an autoencoder is dimensionality reduction and feature discovery. An autoencoder is trained to predict its own input, but to prevent the model from learning the identity mapping, some constraints are applied to the hidden units.[1]

3.2 Denoising Autoencoder (DAE)

Denoising autoencoders are an extension of simple autoencoders. They add noise to inputs during a training process. Autoencoders are one of the unsupervised deep learning models. [1]

Algorithm of DAE:

1. Manually adding noise, for this project, Gaussian Blur.
2. Constructing an autoencoder model. It includes both encoding and decoding layers for compression and decompression respectively.
3. Training the denoising autoencoder model on the noisy dataset. The reconstruction loss between the original image and its reconstruction obtained from final decoding layer is minimised in every epoch.

Figure 3.1 gives an overview of denoising autoencoder.

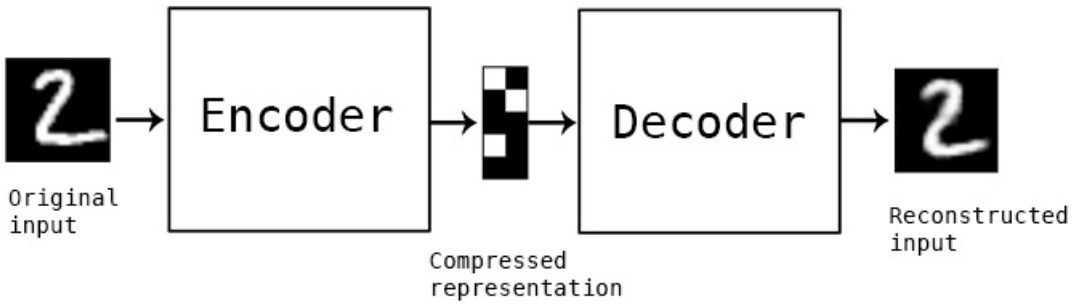


Figure 3.1: Denoising Autoencoder

3.3 Software Setup

1. Keras: Keras is an open-source software library that provides a python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.
2. PyCharm: PyCharm is an integrated development environment used in computer programming, specifically for the Python language.
3. Google Colab: Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

3.4 DAE model summary and training details

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 254, 254, 64)      1792
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 252, 252, 64)      36928
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 250, 250, 32)      18464
_________________________________________________________________
conv2d_transpose (Conv2DTran (None, 252, 252, 32)      9248
_________________________________________________________________
conv2d_transpose_1 (Conv2DTr (None, 254, 254, 64)      18496
_________________________________________________________________
conv2d_transpose_2 (Conv2DTr (None, 256, 256, 64)      36928
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 256, 256, 3)       1731
=================================================================
Total params: 123,587
Trainable params: 123,587
Non-trainable params: 0
_________________________________________________________________
```

Figure 3.2: Model Summary of Denoising Autoencoder

The model is developed and trained using Keras framework with Tensorflow backend. Figure 3.2 gives the structure of the denoising autoencoder model used for deblurring the input image. The model has three convolution layers for encoding or compressing the input images and three convolution transpose layers i.e. convolution + upsampling, for decoding or decompressing the images as shown in figure 3.4. The last convolution layer is used to map the normalised pixel values between 0 and 1. To achieve this, sigmoid activation function is used which is given by 3.1.

$$f(s_i) = \frac{1}{(1 + e^{x})} \tag{3.1}$$

Figure 3.3 shows the sigmoid activation function graph. As x goes to minus infinity, $f(x)$ tends to 0. As x goes to infinity, $f(x)$ tends to 1. At $x = 0$, $f(x) = 0.5$.

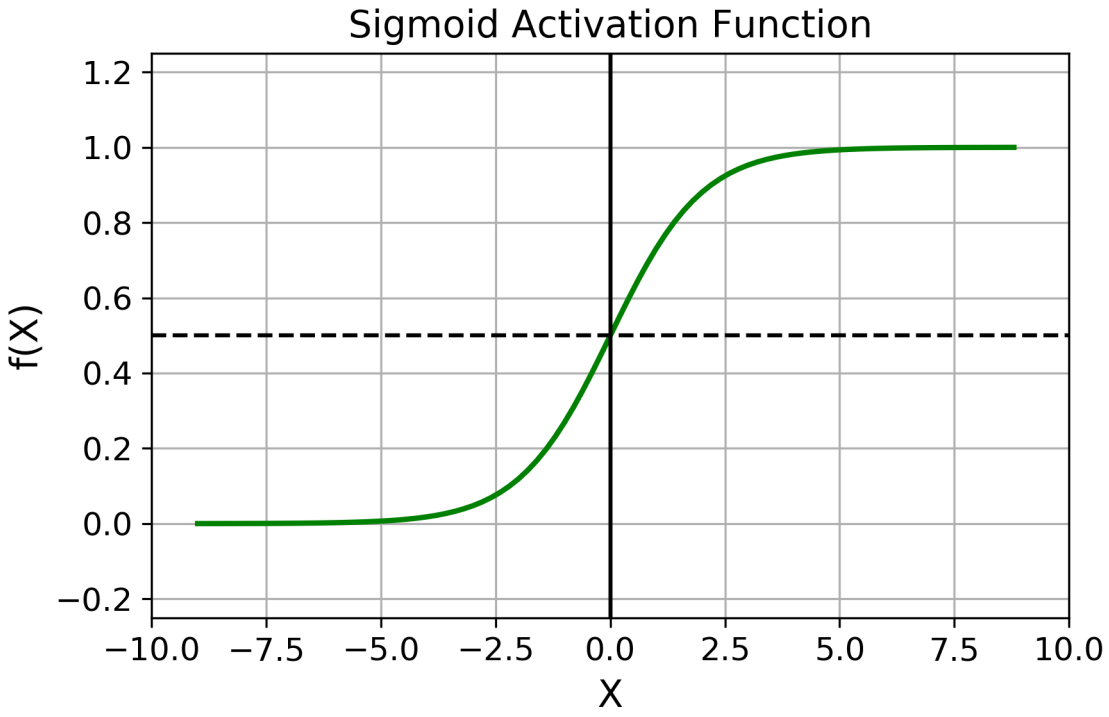


Figure 3.3: Sigmoid activation function

The reconstruction loss is calculated for these mapped values using Binary Cross-Entropy Loss function (Sigmoid Cross-Entropy loss function), which is given by equations 3.2 and 3.3.

$$\begin{aligned} CrossEntropyLoss(CE) &= \sum_{i=1}^{c'=2} (t_i) \log(f(s_i))) \\ &= -(t_1) \log(f(s_1)) - (1 - (t_1)) \log(1 - f(s_1)) \end{aligned} \tag{3.2}$$

$$CE = \begin{cases} -\log(f(s_1)) & if \ t_1 = 1 \\ -\log(1 - f(s_1)) & if \ t_1 = 0 \end{cases} \tag{3.3}$$

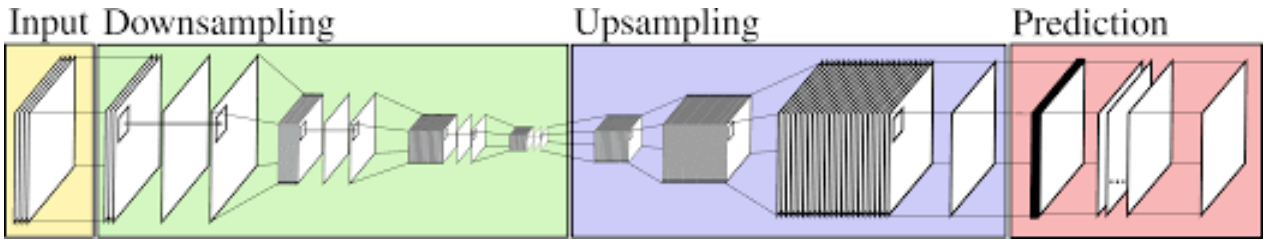


Figure 3.4: Denoising Autoencoder summary

To summarize, a denoising autoencoder is an image processing technique which is

used to remove noise from an image. As shown in figure 3.4, it has both downsampling and upsampling layers. In downsampling the model learns various features of the image, and it reconstructs the image based on the learnt features in upsampling.

Chapter 4

Text Extraction and Summarization

CHAPTER 4

TEXT EXTRACTION AND SUMMARIZATION

Optical Character Recognition (OCR) converts any form of images containing text into digital format. Tesseract module for Python was used to extract the text from the images. The first part of this chapter gives a brief description of text extraction method used in this project and its internal working. Text Summarization is the art of abstracting key content from information sources. Text summarization is one of the many applications of natural language processing and is becoming more popular for information condensation. Natural Language Processing played an important role in developing the summary from a large extract.

4.1 Text extraction

Optical Character Recognition (OCR) is the process of converting any form of text or text-containing documents such as handwritten text, printed or scanned text images, into an editable digital format for deeper and further processing. [6] Tesseract is an open source text recognition (OCR) Engine, available under the Apache 2.0 license. It can be used directly, or (for programmers) using an API to extract printed text from images. It can be used with the existing layout analysis to recognize text within a large document, or it can be used in conjunction with an external text detector to recognize text from an image of a single text line.

Tesseract 3.x was dependant on the multi-stage process:

1. Word finding
2. Line finding
3. Character classification

Word finding is done by organizing text lines into blobs, and the lines and regions are analyzed for fixed pitch or proportional text. Text lines are broken into words differently according to the kind of character spacing. Recognition then proceeds as a two-pass process. In the first pass, an attempt is made to recognize each word. Each word that is satisfactory is passed to an adaptive classifier as training data. The adaptive classifier then recognizes the text accurately.

Modernization of the Tesseract tool was an effort on code cleaning and adding a new LSTM (Long Short Term Memory networks) model. The input image is processed in boxes (rectangle) line by line feeding into the LSTM model and giving output. This algorithm is shown in figure 4.1.

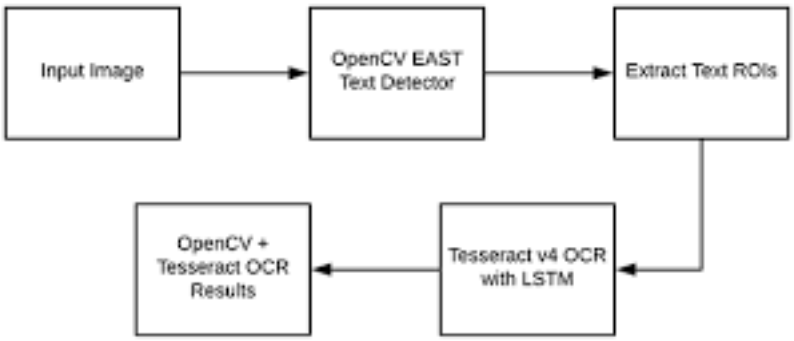


Figure 4.1: Text extraction

4.2 Summarization

Summarization [7] is the process of compressing the original document into a short summary by extracting the most important information from the document. The summary of the document can be helpful to the user to get the main theme of the document in a short span of time. The flow of information in a given document is not uniform, which means that some parts are more important than others. The important task in summarization lies in distinguishing the more informative parts of a document from the less ones.

Extractive vs. Abstractive summarization:

1. An extractive summarization method is the process of selecting most important sentences from the original document

2. An abstractive summarization method produces a short summary by rephrasing the sentences that convey the same information. It involves natural language processing.

Extractive summaries [8] can be formulated by extracting the text segments (sentences or paragraphs) from the text based on the term frequency and sentence similarity.

There are several scenarios where automatic construction of summaries is useful. Other examples include automatic construction of summaries of news articles or email messages for sending them to mobile devices as SMS; summarization of information for government officials, business persons, researches, etc., and summarization of web pages to be shown on the screen of a mobile device, among many others. The Extraction based summarization methods will extracts the most important sentences from the given document. The main challenge of document summarization is to decide which sentences from the input document should be included in a summary. It first, assigns a score to each sentence and then gives ranks to the sentences according to their scores. The sentence with highest score will get the top rank. The score for a sentence is calculated by using statistical features including sentence position, cue words, term frequency, document frequency, topic signature, etc.

Text Summarization is the art of abstracting key content from information sources. Text summarization is one of the many applications of natural language processing and is becoming more popular for information condensation. Natural Language Processing played an important role developing the summary from a large extract. Concepts such as stemming and lemmatization was crucial in the development of the summary. Stemming ensures that all similar words are normalized while lemmatization ensures that infected words were mapped to their healthy dictionary forms. The summarization was done on the basis of eliminating less important sentences from the extract and concurrently keeping the more important ones. The usefulness of a sentence was judged based on the frequency of relevant words in it. The Natural Language Tool Kit (NLTK) of Natural Language Processing (NLP) is used for text summarization.

4.3 Algorithm

Figure 4.2 gives the flow of algorithm for summarization. The algorithm is based on the principle of assigning ranks to sentences based on frequencies of more important words in a given sentence.

1. All stopwords are removed from the text. These include commonly used words such as articles, conjunctions and prepositions. Porter Stemmer, a commonly used

stemmer (and lemmatizer) is implemented using NLTK.

2. A frequency table is constructed to map every word with its frequency in the text.
3. Every sentence is assigned a rank based on the frequency of important words in each sentence.
4. Based on the relative ranks of the sentences, a threshold is decided. Sentences with ranks higher than the threshold are used to make the summary.

By varying the threshold, the depth of detail in the summarized text can also be varied. This is used to provide two stages of summary to the user, a more detailed and a less detailed one.

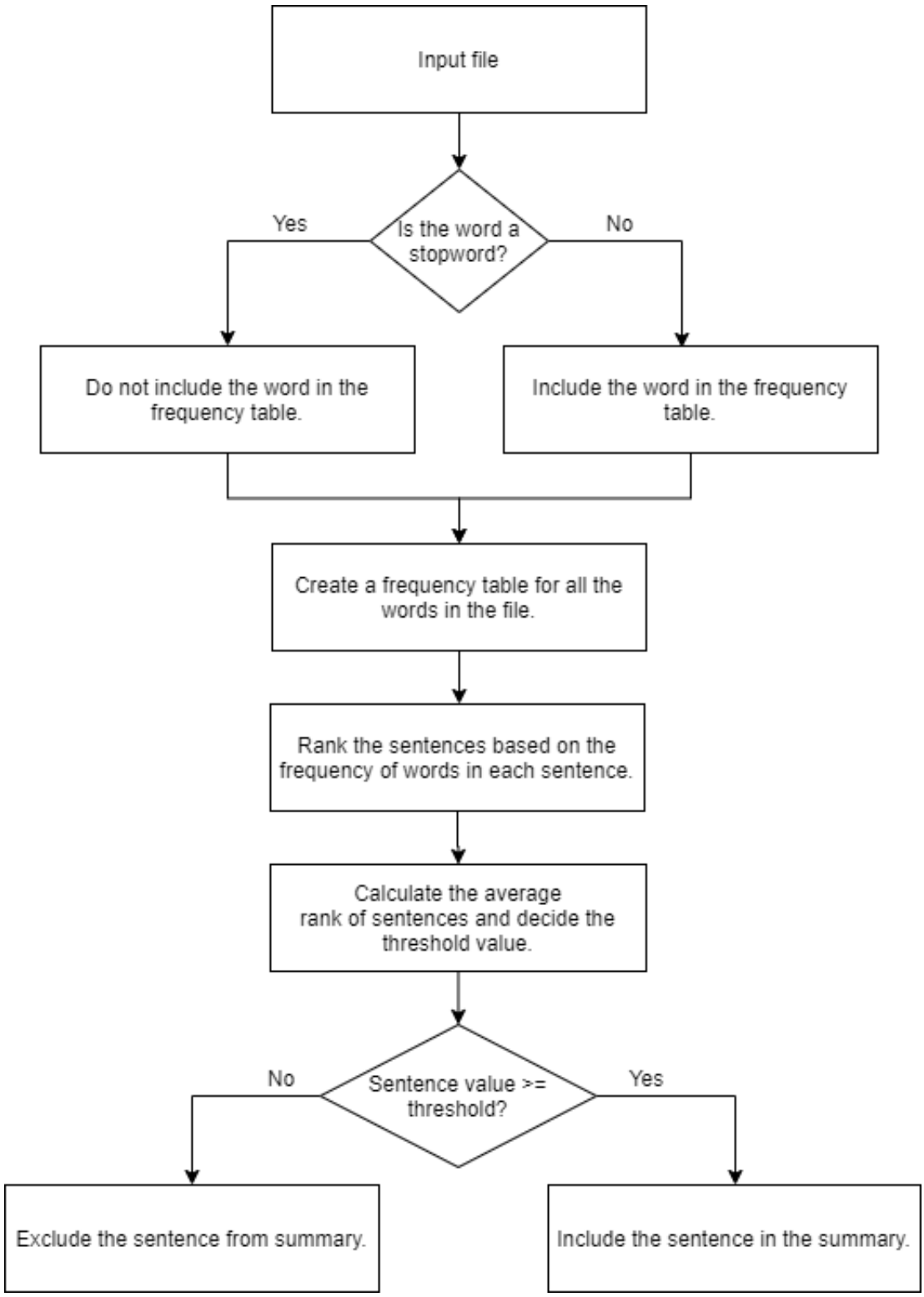


Figure 4.2: Summarization Flow

In summary, text extraction is a process of generating text from electronic documents that have associated text. The tesseract module of python uses OCR to extract text from

images. Text summarization is a method of removing redundant information from text by using frequency tables. Natural language toolkit is used for achieving this.

Chapter 5

Results & Discussions

CHAPTER 5

RESULTS & DISCUSSIONS

The results obtained individually from object detection model, denoising autoencoder and text extraction, and the final result obtained after combining all models with text summarization are included in this chapter. It also includes a text extraction and summarization comparison for pure and deblurred slide images.

5.1 Object detection model

This section includes training details, model accuracy and results obtained for object detection model.

5.1.1 Training details

Table 5.1 gives the training details of object detection model.

Table 5.1: Object detection Training

Number of epochs	16000
Training images used	320
Validation images used	90
Total loss	0.035

5.1.2 Accuracy

Figure 5.1 shows the total loss and figure 5.2 shows the classification loss of object detection model.

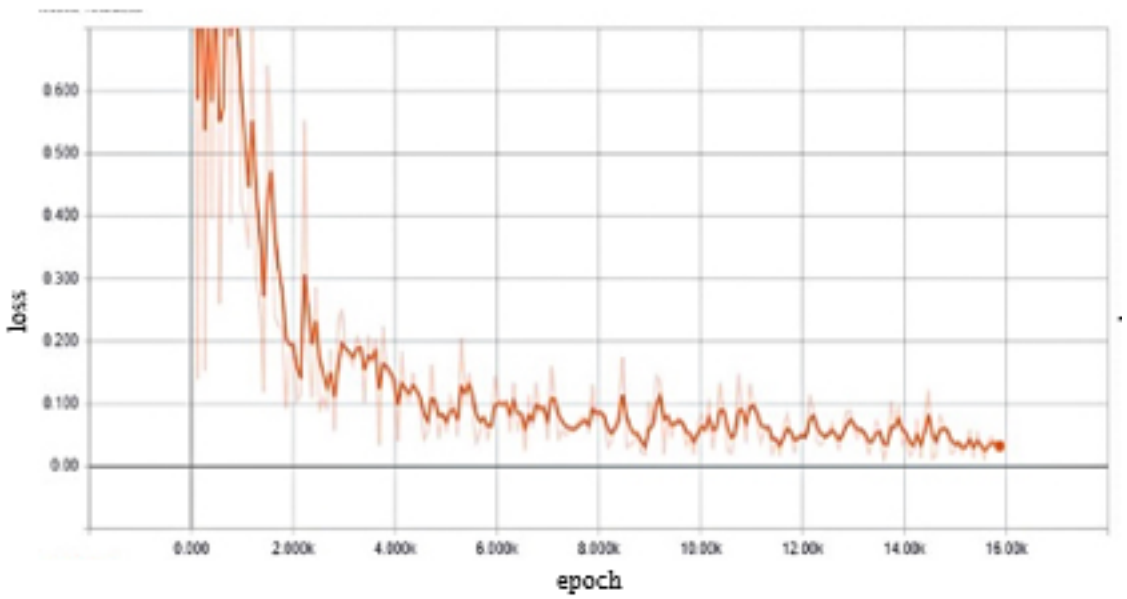


Figure 5.1: Object detection total loss

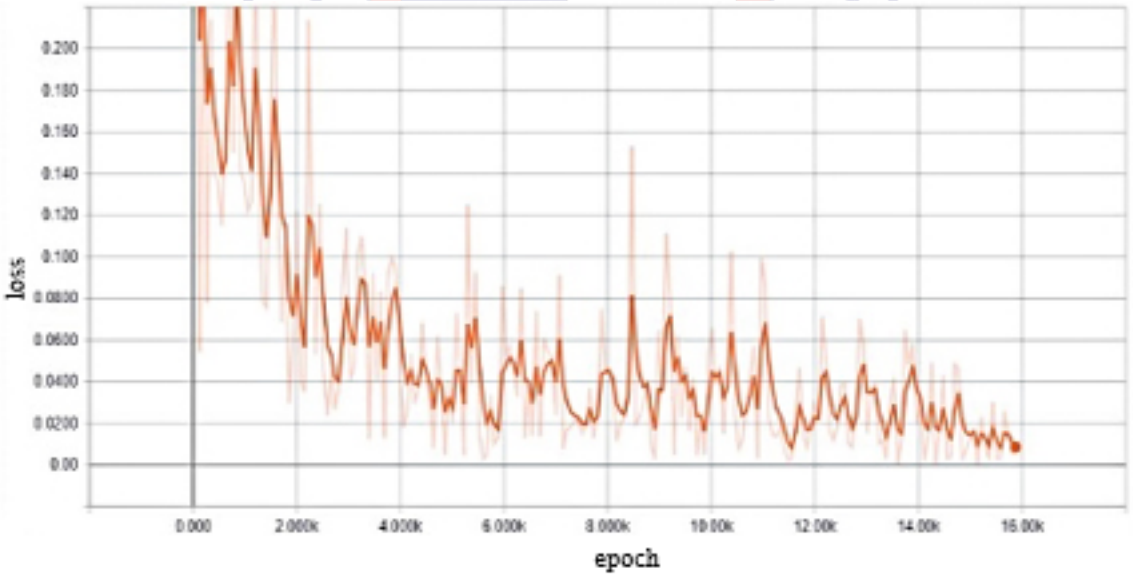


Figure 5.2: Object detection classification loss

5.1.3 Result obtained

Figure 5.3 shows the bounding box created by the object detection model on an input image.

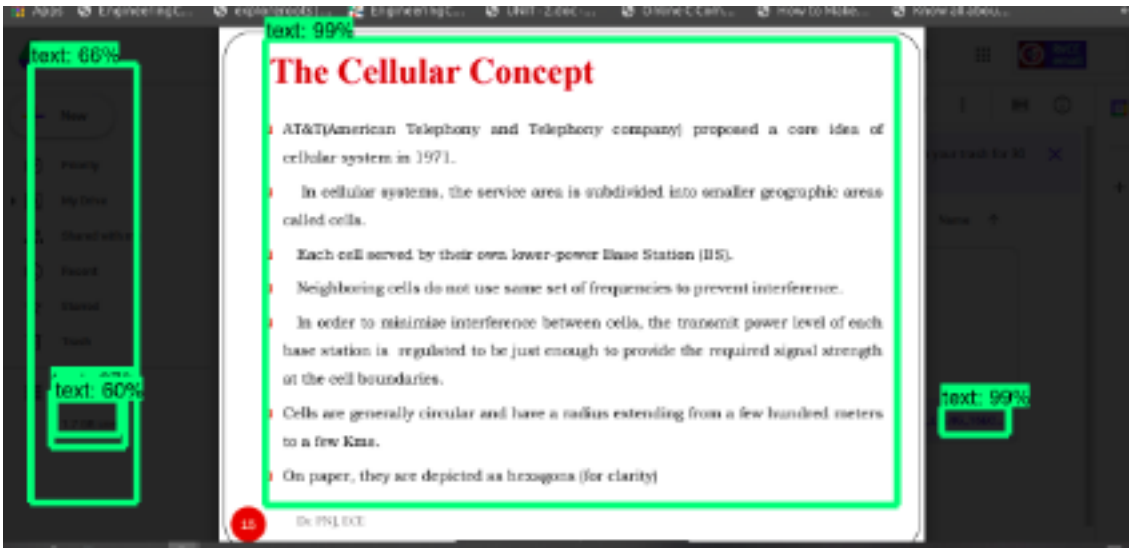


Figure 5.3: Object detection result

5.2 Denoising Autoencoder

This section includes training details, model accuracy and results obtained for denoising autoencoder.

5.2.1 Training details

Table 5.2 gives the training details of denoising autoencoder.

Table 5.2: Denoising Autoencoder training

Number of epochs	1000
Training images used	280
Validation images used	70
Training accuracy	91%
Validation accuracy	87%

5.2.2 Training and validation accuracy

Figure 5.4 shows the graph obtained for training and validation accuracy.

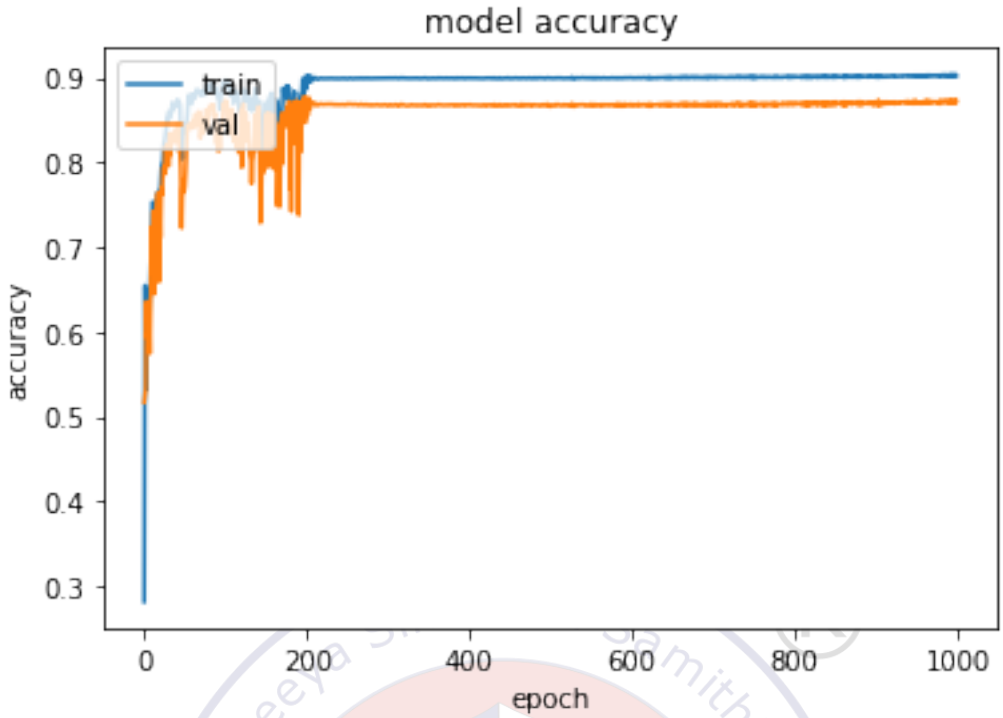


Figure 5.4: Denoising Autoencoder Accuracy

5.2.3 Training and validation loss

Figure 5.5 shows the graph obtained for training and validation loss.

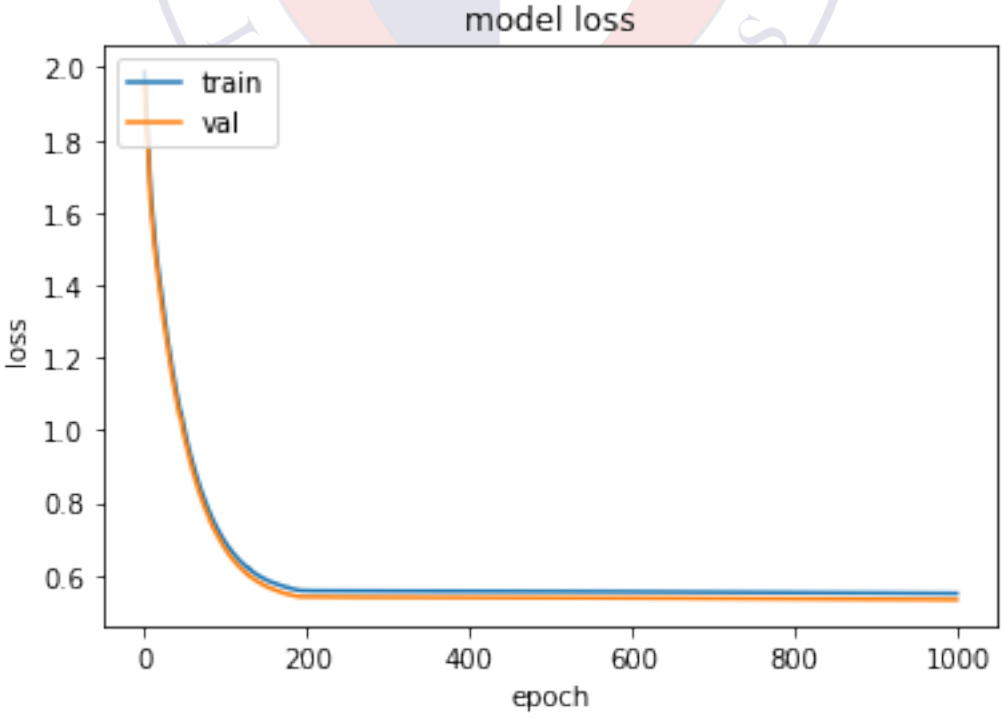


Figure 5.5: Denoising Autoencoder Loss

5.2.4 Result obtained

Figure 5.6 demonstrates the deblurring of an input image by trained denoising autoencoder model.

Figure 5.6: Denoising Autoencoder output

5.3 Text Extraction

This section includes results obtained from text extraction method for a pure, blurred and deblurred image.

5.3.1 Text extraction for pure image

Figure 5.7 shows text detection in a pure slide image and its corresponding text extraction is shown in figure 5.8.

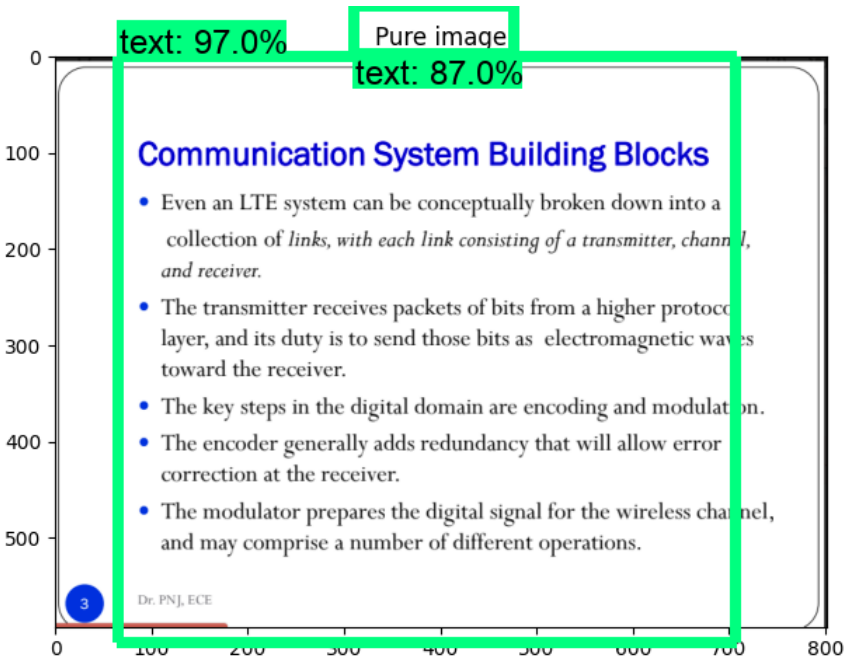


Figure 5.7: Pure image

Pure image

Communication System Building Blocks

* Even an LTE system can be conceptually broken down into a collection of links, with each link consisting of a transmitter, channel, aand receiver.

The transmitter receives packets of bits from a higher protocol layer, and its duty is to send those bits as electromagnetic waves toward the receiver.

The key steps in the digital domain are encoding and modulation.

The encoder generally adds redundancy that will allow error

correction at the receiver.

'The modulator prepares the digital signal for the wireless channel, and may comprise a number of different operations.

Figure 5.8: Text Extracted from pure image

5.3.2 Text extraction for blurred image

Figure 5.9 shows text detection in a blurred slide image and its corresponding text extraction is shown in figure 5.10.

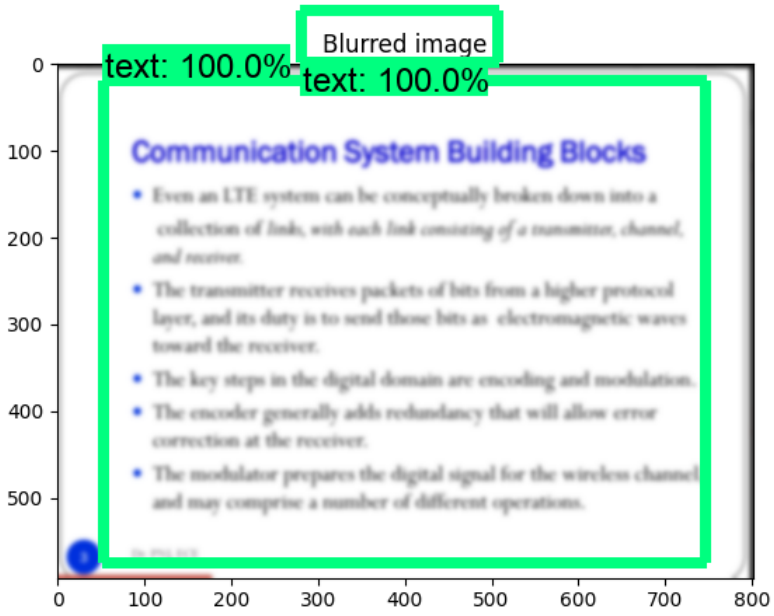


Figure 5.9: Blurred image

Blurred image

Communication System Building Blocks
© Even ae LTE oyster cam be comeptuatly brakes dome ate 4
che te ham oh mh lem ommoctimy of 4 cmamamanen Magma!

nee

9 Thee trasmematter receres pa bets od bets fresem 6 haghery proton od
Lae t arate hat oe ts cete theme bats oe ches treemagme tee anes

fem ard the reserved

0 Thee bey step a the geal deseo are crm satiny are meade

0 The com eater geome ally oth rrchametarm + that oll alle cre
coerce the recener

0 Thee cmchakate pee peares the degtal wegmal bee the are bem «amet
et ems camp es mama td ihe re gue ate

Figure 5.10: Text Extracted from blurred image

5.3.3 Text extraction for deblurred image

Figure 5.11 shows text detection in a deblurred slide image and its corresponding text extraction is shown in figure 5.12.

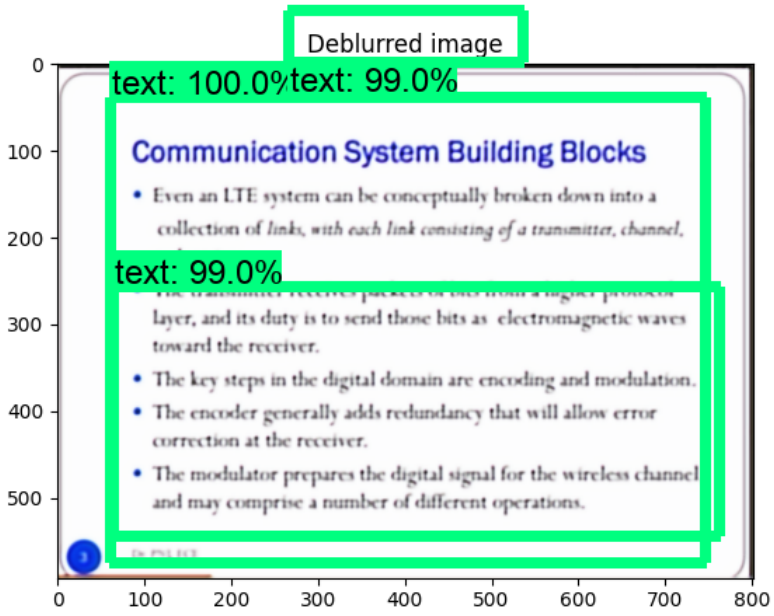


Figure 5.11: Deblurred image

Deblurred image

Communication System Building Blocks

* Even an LTE system can be conceptually broken down into a collection of links, with each link consisting of 4 transmitter, chanel, sand receiver

* The transmitter receives packets of bits from a higher protocol luyer, and its duty ts to send those bits as electromagnetic waves toward the receiver

The key steps tn the digital domain are encoding and modulation,

The encoder generally adds redundancy that will allow error correction at the receiver.

The modulator prepares the digital signal for the wireless channel, and may comprise a number of different operations

Figure 5.12: Text Extracted from deblurred image

5.4 Summarization

5.4.1 Summarization of text extracted from pure slides

Extracted text from pure slides is shown in figure 5.13.

```
ARTIFICIAL INTELLIGENCE
ABSTRACT

¥ It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar
task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.
While no consensual definition of Artificial Intelligence (Al) exists, AI is broadly characterized as the study of computations that allow for
perception, reason and action. This paper examines features of artificial Intelligence, introduction, definitions of AI, history,
applications, growth and achievements.
INTRODUCTION

Artificial Intelligence (AI) is the branch of computer science which deals with intelligence of machines where an intelligent agent is a system
that takes actions which maximize its chances of success. It is the study of ideas which enable computers to do the things that make people
seem intelligent. The central principles of AI include such as reasoning, knowledge, planning, learning, communication, perception and the
ability to move and manipulate objects. It is the science and engineering of making intelligent machines, especially intelligent computer
programs.


ARTIFICIAL INTELLIGENCE
METHODS:

AI methods can be divided into two broad categories: (a) symbolic AI, which focuses on the development of knowledge-based systems (KBS); and
(b) computational intelligence, which includes such methods as neural networks (NN), fuzzy systems (FS), and evolutionary computing. A very
brief introduction to these AI methods is given below, and each method is discussed in more detail in the different sections of this circular.
Knowledge-Based Systems(KBS):
A KBS can be defined as a computer system capable of giving advice in a particular domain, utilizing knowledge provided by a human expert. A
distinguishing feature of KBS lies in the separation behind the knowledge, which can be represented in a number of ways such as rules, frames,
or cases, and the inference engine or algorithm which uses the knowledge base to arrive at a conclusion.
Neural Networks:

NNs are biologically inspired systems consisting of a massively connected network of eSmapHteMOUe “neurons,” organized in layers. By adjusting
the weights of the network, NNs can te “trained” to approximate virtually any nonlinear function to a required degree of accuracy. NNs
typically are rovided with a set of input and output exemplars. A earning algorithm (such as back propagation) would then be used to adjust
the weights in the network so that the network would give ifs desired output, in a type of learning commonly called supervised learning.




Definitions of AI

Computers with the ability to mimic or duplicate the functions of the human brain.
# Artificial Intelligence (AI) is the study of how computer systems can simulate intelligent processes such as learning, reasoning, and
understanding symbolic information in context. = It is the science and engineering of making intelligent machines, especially intelligent
computer programs. It is related to the similar task of using computers to understand human intelligence. = “The branch of computer science
that is concerned with the automation of intelligent behaviour” (Luger and Stubblefield, 1993).
History
The modern history of AI can be traced back to the year 1956 when John McCarthy proposed the term as the topic for a conference held at
Dartmouth College.
History(Contd.)

The initial goals for the field were too ambitious and the first few AI systems failed to deliver what was promised. After a few of these
early failures, AI researchers started setting some more realistic goals for themselves. In the 1960s and the 1970s, the focus of AI research
was primarily on the development of KBS or expert systems.
History(contd.)

The late 1980s and the 1990s saw a renewed interest in NN research when several different researchers reinvented the back propagation learning
algorithm (although the algorithm was really first discovered in 1569). The back faepaeaunn algorithm was soon applied to man: learning
problems causing great excitement within the AI community. There is also a move toward the development of hybrid intelligent systems (i.¢.,
systems that use more than one AI method)
```

Figure 5.13: Text extracted from pure slides

Summary generated for the above text is shown in figure 5.14.

It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable. While no consensual definition of Artificial Intelligence (Al) exists, AI is broadly characterized as the study of computations that allow for perception, reason and action. This paper examines features of artificial Intelligence, introduction, definitions of AI, history, applications, growth and achievements.
INTRODUCTION

Artificial Intelligence (AI) is the
branch of computer science which deals with intelligence of machines where an intelligent agent is a system that takes actions which maximize its chances of success. The central principles of AI include such as reasoning, knowledge, planning, learning, communication, perception and the ability to move and manipulate objects.
ARTIFICIAL INTELLIGENCE METHODS:

AI methods can be divided into two broad categories: (a) symbolic AI, which focuses on the development of knowledge-based systems (KBS); and (b) computational intelligence, which includes such methods as neural networks (NN), fuzzy systems (FS), and evolutionary computing. A very brief introduction to these AI methods is given below, and each method is discussed in more detail in the different sections of this circular. Knowledge-Based Systems(KBS): A KBS can be defined as a computer system capable of giving advice in a particular domain, utilizing knowledge provided by a human expert. Neural Networks: NNs are biologically inspired systems consisting of a massively connected network of eSmapHteMOUe “neurons,” organized in layers. A earning algorithm (such as back propagation) would then be used to adjust the weights in the network so that the network would give ifs desired output, in a type of learning commonly called supervised learning. # Artificial Intelligence (AI) is the study of how computer systems can simulate intelligent processes such as learning, reasoning, and understanding symbolic information in context. = “The branch of computer science that is concerned with the automation of intelligent behaviour” (Luger and Stubblefield, 1993). After a few of these early failures, AI researchers started setting some more realistic goals for themselves. In the 1960s and the 1970s, the focus of AI research was primarily on the development of KBS or expert systems. There is also a move toward the development of hybrid intelligent systems (i.ç., systems that use more than one AI method)

Figure 5.14: Text summarized from pure slides

5.4.2 Summarization of text extracted from deblurred slides

Extracted text from deblurred slides is shown in figure 5.15.

ABSTRACT

It is the sctence and engineering of maling intelligent machines, expecially intelligent compater programs. It is related to the similar task of using Computers to understand human intelligence, but Al does not have to confine itelf to methods that are Molegically observable. While no consenrmal definition of Artificial © (AD exists, Al it broadly characterized as the s of computations that allow for perception, reason and action This paper cramines features of artificial latelligence. introduction, definitions of Al. history, applications. growth and achievements.
INTRODUCTION
Artificial Intelligence (Al) ts the branch of co science which deals with igenee of machines where an intelligent agent ts a system that takes actions whach matimize its chances of success. 11 is the study of kdeas whoch cnable computers to do the things poet make people secm inte! The central principles of Al bechode such as reasoning, Loo ledge, planning. : fetralog: Communication, Perecpoen and the ability to move and ite © ts. It ts the science and aa Tt eakiog intellogent machines, especially intelligent Computer programs.

ARTIFICIAL INTELLIGENCE
METHODS:
Al methods can be divided into two broad categories: (a) symbolic Al, which focuses on the development of knowledge-based systems (KBS); and (+) computational intelligence, which includes such methods as neural networks (NN), fuzzy systems (FS), and evolutionary computing A very brief introduction to these Al methods is given below, and cach method is discussed in more detail in the different sections of this circular.
Knowledge-Based Systeme KBS):
A KBS can be defined as a computer system capable of giving advice ina ear domain, utilizing knowledge provided by a human expert. A distinguishing feature of KBS lies in the separation behind the knowledge, which can be represented in a number of ways such as mules, frames, or cases, and the inference engine or algorithm which uses the knowledge base to amtive at a conclusion
Neural Networks: NNs are biologically inspired systems eee ofa massively estes network of oasis “neurons, Pied ocnes in layers. B Pestin the weights of the network, NNs tan be “tra to approximate virtually any pbaliocar function to a reyes degree of accuracy. NNs typically are with a set of input and output exemplars. A Fanon algorithm (such as back propagation) would then be used to adjust the wei; Fes ibe erator ae that the network would give the desired output, in a type of learning commonly called supervised learning.
Definitions of Al

Computers with the ability to mimic or duplicate the functions of the human brain.
Antificial Intelligence (Al) is the study of how computer systems can simulate intelligent processes such as Ieaming, reasoning, and understanding symbolic information in context. It is the science and enginecring of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence.
© “The branch of computer science that is concerned with the automation of intelligent behaviour” (Luger and Stubblefield, 1993).

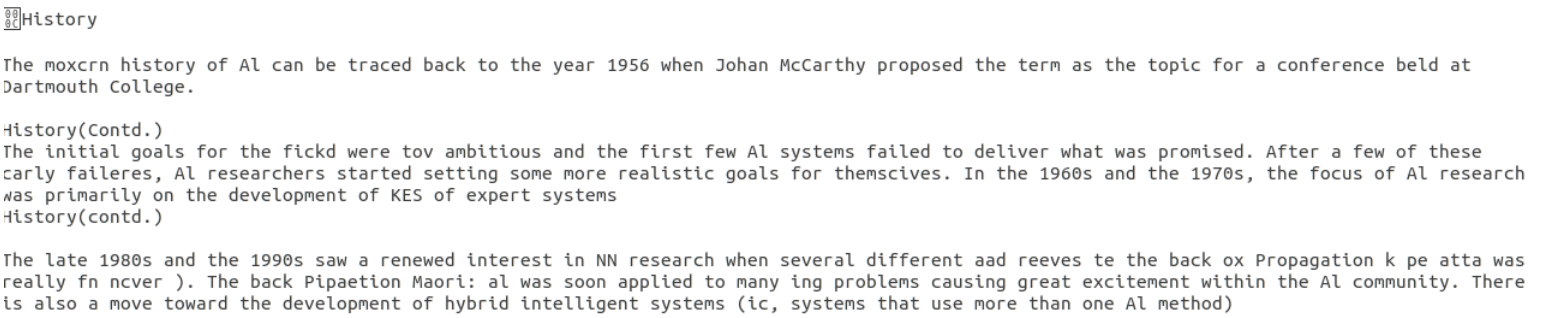


Figure 5.15: Text extracted from deblurred slides

Summary generated for the above text is shown in figure 5.16.

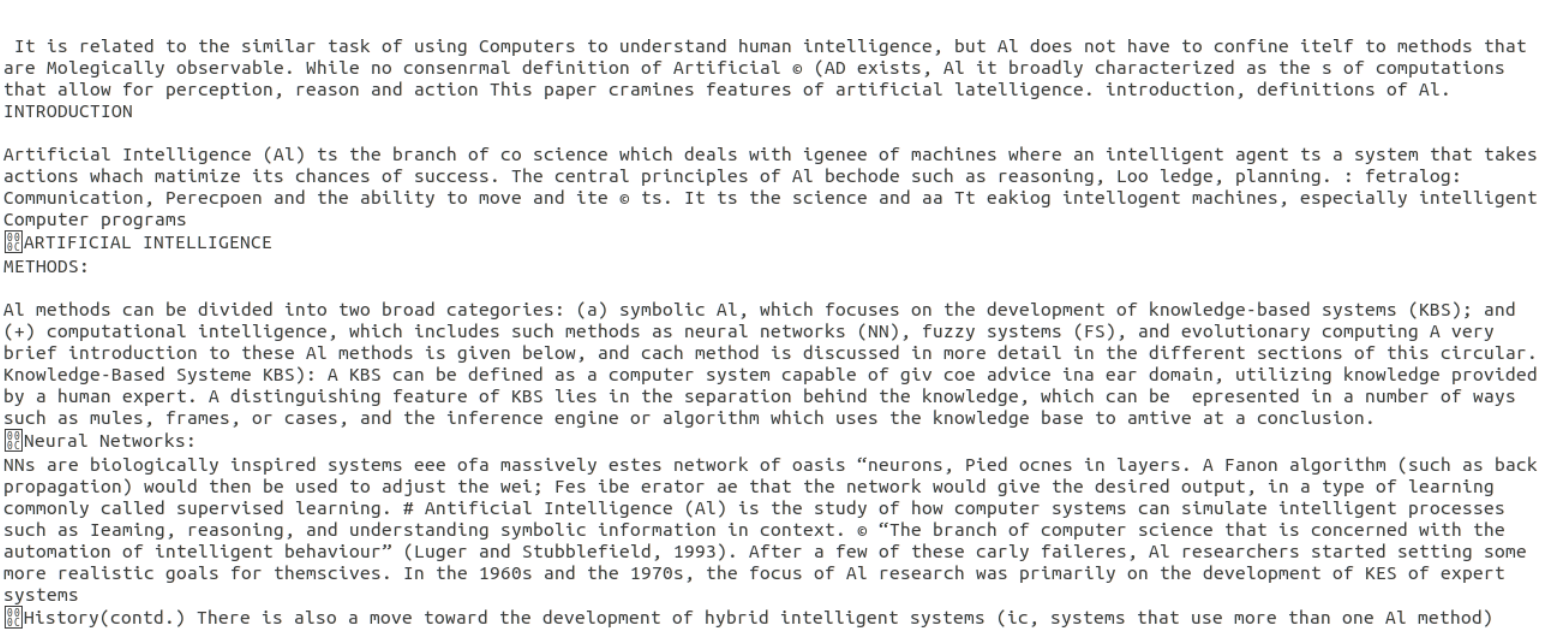


Figure 5.16: Text summarized from deblurred slides

To summarize, this chapter gives independent results obtained for each trained model. Final section gives the result obtained after combining all the models. It also gives a comparison of results obtained for pure, blurred and deblurred images where deblurring is achieved using denoising autoencoder.

Chapter 6

Conclusion and Future Scope

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

This project provides a smart solution of summarizing the entire presentation and logging it. There are two main objectives of this project. The first objective of this project is to build a more efficient object detection model compared to OCR (Optical Character Recognition) since OCR has many drawbacks associated with it such as reduced accuracy when input images are skewed or have low lighting. The second one is to extract text from blurred images and summarize the same.

The first objective was met by implementing a faster R–CNN model for text detection. This model is proven to have better accuracy compared to OCR. The second objective was met by developing a denoising autoencoder which will deblur an image before passing it through the object detection model. This ensures better extraction efficiency.

An efficient faster R–CNN model was developed for text detection which yielded a total loss of only 0.035. Denoising autoencoder was successfully developed and trained to obtain a training accuracy of 91% and a validation accuracy of up to 87%. This accuracy can be further improved by training the model on a larger dataset for more number of epochs.

6.2 Future Scope

The future scope of the project should not limit its usage to classrooms or small presentations, but should be able to extend its capability, to cater to larger presentations and seminars. In order to achieve better quality text detection, an area that can be addressed would be recording the audio and integrating the audio and images to get a detailed summary. Noise may persist in the case of lower-grade equipment or even improper placement of the device on uneven terrain. Noise detection can improve the scope of use of this product. The microphone is extremely susceptible to external interference from the surroundings. To address this issue, better-grade equipment can be installed in the audio interface. Another solution is to ensure that the speaker, or the person making the delivery has a personalized microphone specifically for this purpose.

Another extension that can be made is to generate personalized summaries specific to

each user. A user who has paid more attention to the delivery or has prior knowledge of the topic, may not need as detailed a summary as to someone who has paid less attention or someone who is not well–versed in that particular field. The personalization can be used to cater to these specific needs.

6.3 Learning Outcomes of the Project

Over the course of the project, many concepts and tools were learnt, and the project was built successfully.

- Domains like Image Processing and machine learning, were explored in good depth in this project.
- The working of CNNs and extend its implementation to object detection model and denoising autoencoder were understood.
- Identifying hardware limitations for training the models and overcoming the problems that they might cause by implementing optimization methods like L2 regularization was accomplished.
- The concepts of Natural Language Processing were understood, and then applied successfully to get the summary of the extracted text.
- Shortcomings for industrial applicability were identified and some of those issues were addressed in this report.

BIBLIOGRAPHY

[1] L. Yasenko, Y. Klyatchenko, and O. Tarasenko-Klyatchenko, "Image noise reduction by denoising autoencoder," in *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, IEEE, 2020, pp. 351–355.

[2] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019. DOI: 10.1109/TNNLS.2018.2876865.

[3] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128 837–128 868, 2019.

[4] A. Bhat, A. C. Rao, A. Bhaskar, V. Adithya, and D. Pratiba, "A cost-effective audio-visual summarizer for summarization of presentations and seminars," in *2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)*, IEEE, 2018, pp. 271–276.

[5] B. Wang, J. Xu, J. Li, C. Hu, and J.-S. Pan, "Scene text recognition algorithm based on faster rcnn," in *2017 First International Conference on Electronics Instrumentation & Information Systems (EIIS)*, IEEE, 2017, pp. 1–4.

[6] K. A. Hamad and M. Kaya, "A detailed analysis of optical character recognition technology," *International Journal of Applied Mathematics, Electronics and Computers*, vol. 4, no. 1, pp. 244–249, 2016.

[7] Y. S. Reddy and A. S. Kumar, "An efficient approach for web document summarization by sentence ranking," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 7, 2012.

[8] J. Madhuri and R. G. Kumar, "Extractive text summarization using sentence ranking," in *2019 International Conference on Data Science and Communication (IconDSC)*, IEEE, 2019, pp. 1–3.