VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

J-COMPONENT, November 2019

B.Tech, Fall-2019-2020

TITLE	**Real Time Object Detection with OpenCV and MobileNetSSD**
COURSE CODE	ITE1015
COURSE NAME	Soft Computing
SLOT	E2+TE2
FACULTY	Prof. Brijendra Singh

TEAM 7:

Group Members Name	Registration No.
Nitin Vankadari	18BIT0227
Priyal Bhardwaj	18BIT0272
Rohan Jain	18BIT0429

CONTENT

LITERATURE REVIEW

Paper 1:
Object Detection with Deep Learning

This paper deals with the field of computer vision, mainly for the application of deep learning in object detection task. On the one hand, there is a simple summary of the datasets and deep learning algorithms commonly used in computer vision. On the other hand, a new dataset is built according to those commonly used datasets, and choose one of the networks called faster R-CNN to work on this new dataset. Through the experiment to strengthen the understanding of these networks, and through the analysis of the results learn the importance of deep learning technology, and the importance of the dataset for deep learning.

--- Xinyi Zhou1, Wei Gong2, WenLong Fu3, Fengtong Du, Information Engineering School, Communication University of China,CUC, Neuroscience and Intelligent Media Institute, Communication University of China Beijng, China

Paper 2:
Object Detection using Open CV - Python

Object detection is a well-known computer technology connected with computer vision and image processing that focuses on detecting objects or its instances of a certain class (such as humans, flowers, animals) in digital images and videos. There are various applications of object detection that have been well researched including face detection, character recognition, and vehicle calculator. Object detection can be used for various purposes including retrieval and surveillance. In this study, various basic concepts used in object detection while making use of OpenCV library of python 2.7, improving the efficiency and accuracy of object detection are presented.

--- Bhumika Gupta, PhD Assistant Professor, C.S.E.D, Ashish Chaube B.Tech IV Year, Ashish Negi B.Tech IV Year, Umang Goel B.Tech IV Year

Paper 3:
YOLOv2

We introduce YOLO, a unified model for object detection. Our model is simple to construct and can be trained directly on full images. Unlike classifier-based approaches, YOLO is trained on a loss function that directly corresponds to detection performance and the entire model is trained jointly. Fast YOLO is the fastest general-purpose object detector in the literature and YOLO pushes the state-of-the-art in real-time object detection. YOLO also generalizes well to new domains making it ideal for applications that rely on fast, robust object detection.

--- Joseph Redmon∗, Santosh Divvala∗†, Ross Girshick¶, Ali Farhadi∗†
University of Washington∗, Allen Institute for AI†, Facebook AI Research¶

Paper 4:
Deep Learning for Real-Time Capable Object Detection and Localization on Mobile Platforms

In order to interact with humans, the platforms need an in-depth knowledge of the environment. Hence, it is required to detect a variety of static and non-static objects. Goal of this paper is to propose an accurate and real-time capable object detection and localization approach for the use on mobile platforms. A method is introduced to use the powerful detection capabilities of a neural network for the localization of objects. Therefore, detection information of a neural network is combined with depth information from a RGB-D camera, which is mounted on a mobile platform. As detection network, YOLO Version 2 (YOLOv2) is used on a mobile robot.

--- F. Particke1, *, R. Kolbenschlag1, M. Hiller1, L. Patiño-Studencki1 and J. Thielecke, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Information Technologies, Erlangen, Gersmany

Paper 5:
Deep Learning Towards Mobile Applications

There exist many challenges to realize deep learning in mobile applications, including the contradiction between the miniature nature of mobile devices and the resource requirement of deep neural networks, the privacy and security concerns about individuals' data, and so on. To resolve these challenges, during the past few years, great leaps have been made in this area. In this paper, we provide an overview of the current challenges and representative achievements about pushing deep learning on mobile devices from three aspects: training with mobile data, efficient inference on mobile devices, and applications of mobile deep learning.

--- Ji Wang∗, Bokai Cao†, Philip S. Yu†‡, Lichao Sun†, Weidong Bao∗, and Xiaomin Zhu∗ ∗College of Systems Engineering, National University of Defense Technology, Changsha, Hunan, P. R. China †Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA ‡Institute for Data Science, Tsinghua University, Beijing, P. R. China

Paper 6:
Deep Learning

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

-- Krizhevsky, A., Sutskever, I. & Hinton, G. ImageNet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems 25* 1090–1098 (2012).

Paper 7:
An Overview of Deep Learning-Based Object Detection Methods

In recent years, there has been rapid development in the research area of deep learning. Deep learning was used to solve different problems, such as visual recognition, speech recognition and handwriting recognition and was achieved a very good performance. In deep learning, Convolutional Neural Networks are found to give the most accurate results, in solving object detection problems. In this paper there is summarizing some of the most important deep learning

models used for objects detection tasks over these last recent years. Then, there is comparison in terms of speed and accuracy between the most used state-of-the-arts methods in object detection.
-- P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, pp. 1627-1645, 2010.

Paper 8:
Deep convolutional neural network based species recognition for wild animal monitoring

We proposed a novel deep convolutional neural network based species recognition algorithm for wild animal classification on very challenging camera-trap imagery data. The imagery data were captured with motion triggered camera trap and were segmented automatically using the state-of-the-art graph-cut algorithm. The moving foreground is selected as the region of interests and is fed to the proposed species recognition algorithm. For the comparison purpose, we use the traditional bag of visual words model as the baseline species recognition algorithm. It is clear that the proposed deep convolutional neural network based species recognition achieves superior performance. To our best knowledge, this is the first attempt to the fully automatic computer vision based species recognition on the real camera-trap images.

-- Guobin Chen, Tony X. Han, Zhihai He∗ Univeristy of Missouri Electitral and Comptuer Engineering Department Columbia, MO 65203, USA *Roland Kays, and Tavis Forrester* North Carolina State University Department of Forestry and Environmental Resources Raleigh, NC 27607, USA

Paper 9:
Automatically identifying wild animals in camera trap images with deep learning

Having accurate, detailed, and up-to-date information about wildlife location and behavior across broad geographic areas would revolutionize our ability to study, conserve, and manage species and ecosystems. Currently such data are mostly gathered manually at great expense, and thus are sparsely and infrequently collected. Here we investigate the ability to automatically, accurately, and inexpensively collect such data from motion sensor cameras.

These camera traps enable pictures of wildlife to be collected inexpensively, and at high-volume. However, identifying the animals, animal attributes, and behaviours in these pictures remains an expensive, time-consuming, manual task often performed by researchers, hired technicians. In this paper, demonstrate that such data can be automatically extracted by deep neural networks (aka deep learning), which is a cutting-edge type of artificial intelligence.

-- Mohammed Sadegh Norouzzadeh1, Anh Nguyen1, Margaret Kosmala2, Ali Swanson3, Craig Packer4, and Jeff Clune1,5 1University of Wyoming; 2Harvard University; 3University of Oxford; 4University of Minnesota; 5Uber AI Labs

Paper 10:
Fast object detection in compressed JPEG Images

Object detection in still images has drawn a lot of attention over past few years, and with the advent of Deep Learning impressive performances have been achieved with numerous industrial applications. Most of these deep learning models rely on RGB images to localize and identify objects in the image. However, in some application scenario, images are compressed either for storage savings or fast transmission. Therefore, a time-consuming image decompression step is compulsory in order to apply the aforementioned deep models. To alleviate this drawback, we propose a fast, deep architecture for object detection in JPEG images, one of the most widespread compression formats.

-- Benjamin Deguerre1,2, Clement Chatelain´ 1, Gilles Gasso1

Paper 11:
Vehicle target detection in complex scenes based on YOLOv3 algorithm

In view of the low accuracy of traditional vehicle target detection methods in complex scenes, combined with the current hot development of deep learning, this paper applies the YOLOv3 algorithm framework to achieve vehicle target detection. By using PASCAL VOC2007 and VOC2012 data sets, images containing vehicle targets were screened out to constitute the VOC car data set, and the target detection problem was converted into a binary classification problem. Then loading the pre-trained YOLOv3 model weight, and training the vehicle target detection model weight based on YOLOv3 algorithm, which is used to detect the test samples. Experimental results show that this method has advantages over the traditional target detection algorithms in recognition accuracy and detection speed.

-- Lecheng Ouyang1, Huali Wang1*
1 College of Communication Engineering, PLA Army Engineering University, Nanjing, Jiangsu, 210007, China

Paper 12:
YOLO: Unified, Real-Time Object Detection

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to per-form detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. Our unified architecture is extremely fast.

--Joseph Redmon∗, Santosh Divvala∗†, Ross Girshick¶, Ali Farhadi∗†
University of Washington∗, Allen Institute for AI†, Facebook AI Research¶

INTRODUCTION

Object Detection is the process of finding real-world object instances like car, bike, TV, flowers, and humans in still images or Videos. It allows for the recognition, localization, and detection of multiple objects within an image which provides us with a much better understanding of an image as a whole. Object Detection can be done via multiple ways one of which happens to be Deep Learning.

Efficient and accurate object detection has been an important topic in the advancement of computer vision systems. With the advancement of deep learning techniques, the accuracy for object detection has increased a lot. Our project aims to incorporate efficient techniques for object detection with the goal of achieving high speed with a real-time performance.

In our project, we have used a completely deep learning-based approach to solve the problem of object. The resulting system is fast helps those real-time applications which require object detection.

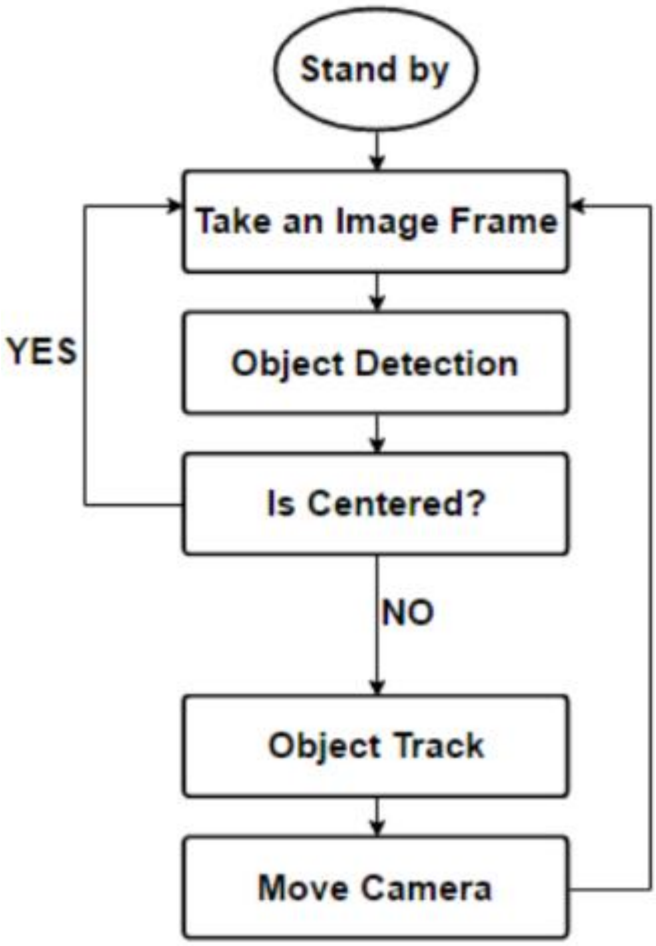
PROPOSED METHODOLOGY

Normally for Image recognition a convolution neural network is used where the data is image. While in an RCNN the R standing for region is for object detection. A typical CNN can only tell you the class of the object but not where they are located.

In the case of YOLO which is used for real time object detection, YOU ONLY LOOK ONCE is a system for detection objects in the PASCAL VOC 2012 dataset. It can detect the 20 Pascal objects.

But in our proposed methodology instead of RCNN and YOLO we are using MobileNet SSD.

SSD is Single Shot multi-box Detector for real time processing. RCNN creates boundary boxes then uses them to classify objects, but SSD uses default boxes and has multi-scale features and also use lower resolution images so the speed is higher which makes it faster than RCNN.



IMPLEMENTATION / CODE

```
# USAGE
# python real_time_object_detection.py --prototxt
MobileNetSSD_deploy.prototxt.txt --model
MobileNetSSD_deploy.caffemodel

# import the necessary packages
from imutils.video import VideoStream
from imutils.video import FPS
import numpy as np
import argparse
import imutils
import time
import cv2

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--prototxt", required=True,
help="path to Caffe 'deploy' prototxt file")
ap.add_argument("-m", "--model", required=True,
help="path to Caffe pre-trained model")
ap.add_argument("-c", "--confidence", type=float, default=0.2,
help="minimum probability to filter weak detections")
args = vars(ap.parse_args())

# initialize the list of class labels MobileNet SSD was trained to
# detect, then generate a set of bounding box colors for each class
CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
"bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
"dog", "horse", "motorbike", "person", "pottedplant", "sheep",
"sofa", "train", "tvmonitor","weapon"]
COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))

# load our serialized model from disk
print("[INFO] loading model...")
net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])
```

```
# initialize the video stream, allow the cammera sensor to warmup,
# and initialize the FPS counter
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
time.sleep(2.0)
fps = FPS().start()

# loop over the frames from the video stream
while True:
# grab the frame from the threaded video stream and resize it
# to have a maximum width of 400 pixels
frame = vs.read()
frame = imutils.resize(frame, width=400)

# grab the frame dimensions and convert it to a blob
(h, w) = frame.shape[:2]
blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)),
0.007843, (300, 300), 127.5)

# pass the blob through the network and obtain the detections and
# predictions
net.setInput(blob)
detections = net.forward()

# loop over the detections
for i in np.arange(0, detections.shape[2]):
# extract the confidence (i.e., probability) associated with
# the prediction
confidence = detections[0, 0, i, 2]

# filter out weak detections by ensuring the `confidence` is
# greater than the minimum confidence
if confidence > args["confidence"]:
# extract the index of the class label from the
# `detections`, then compute the (x, y)-coordinates of
# the bounding box for the object
idx = int(detections[0, 0, i, 1])
```

```
box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
(startX, startY, endX, endY) = box.astype("int")

# draw the prediction on the frame
label = "{}: {:.2f}%".format(CLASSES[idx],
confidence * 100)
cv2.rectangle(frame, (startX, startY), (endX, endY),
COLORS[idx], 2)
y = startY - 15 if startY - 15 > 15 else startY + 15
cv2.putText(frame, label, (startX, y),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[idx], 2)

# show the output frame
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF

# if the `q` key was pressed, break from the loop
if key == ord("q"):
break

# update the FPS counter
fps.update()

# stop the timer and display FPS information
fps.stop()
print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))
print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))

# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()
```

TOOLS USED

1) Python: Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.

2) OpenCV: Open Source Computer Vision Library is an open source computer vision and machine learning software library.

3) NumPy: NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

4) MobileNet: MobileNet is an architecture which is more suitable for mobile and embedded based vision applications where there is lack of compute power.

5) SSD: Single Shot MultiBox Detector for real-time processing speeds up the process of object detection by eliminating the need of the region proposal network.

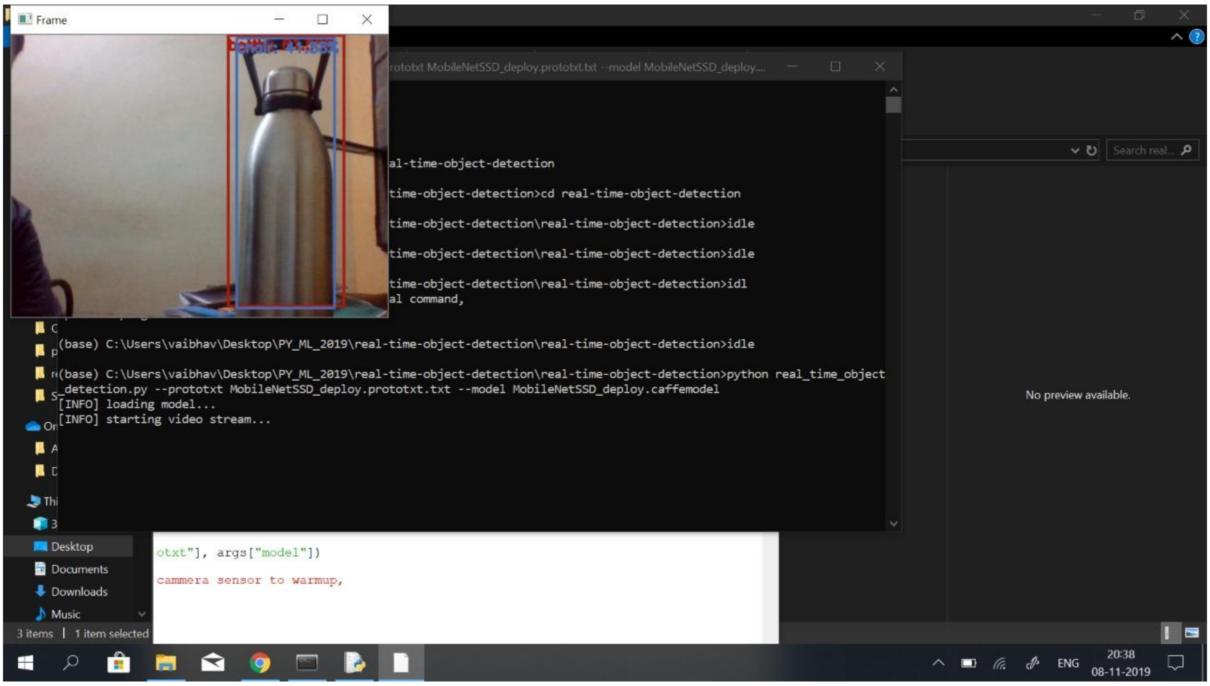
6) CV2: It is the latest OpenCV interface in which everything is returned as NumPy objects like ndarray and native Python objects like lists, tuples, dictionary, etc. So due to this NumPy support, you can do any numpy operation here.

7) COCO Dataset: COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features like object segmentation, recognition in context etc.

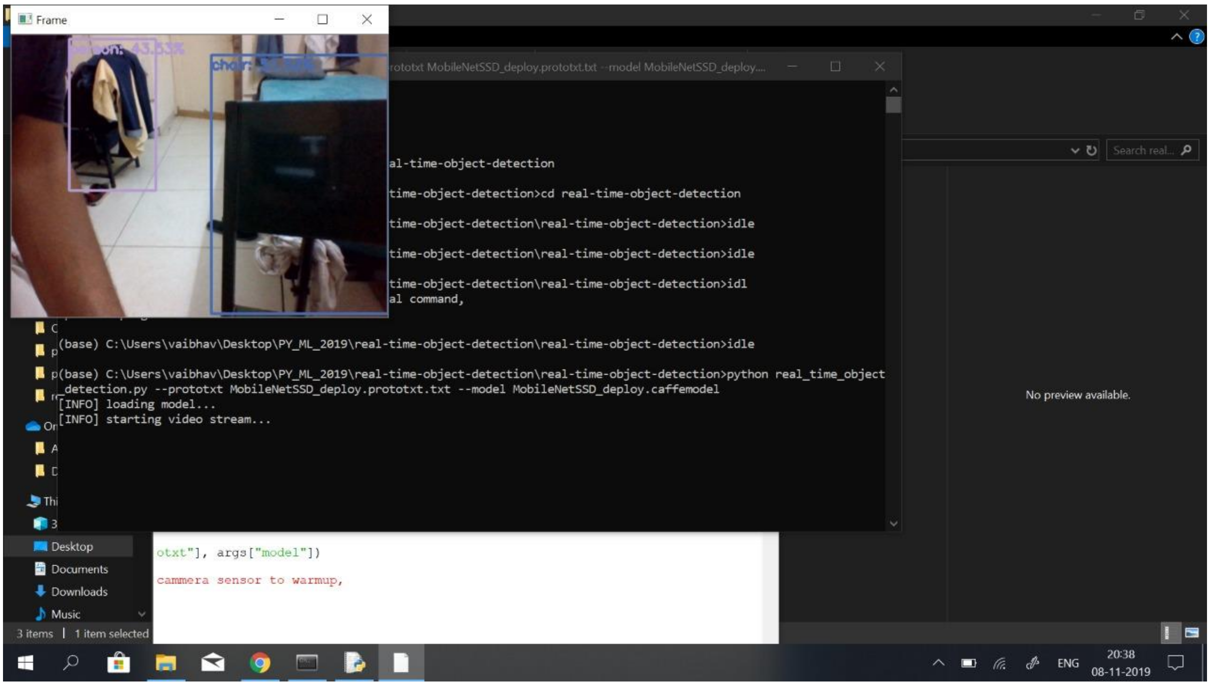
8) CAFFE: Convolutional Architecture for Fast Feature Embedding is a deep learning framework. It is open source, under a BSD license. It is written in C++, with a Python interface.

SNAPSHOTS

Detecting Bottle:



Detecting Chair:



FUTURE WORK/ENHANCEMENTS

An important scope would be to train the system on a video sequence for usage in tracking applications. Addition of a temporally consistent network would enable smooth detection and more optimal than per-frame detection.

Emergency alert systems serve as a critical link in the chain of crisis communication, and they are essential to minimize loss during emergencies. Acts of terrorism and violence, chemical spills, amber alerts, nuclear facility problems, weather-related emergencies, flu pandemics, and other emergencies all require those responsible such as government officials, building managers, and university administrators to be able to quickly and reliably distribute emergency information to the public. Linking the object detection system to a buzzer will immediately alert authorities if a dangerous item like gun is detected.

CONCLUSION

Real time object detection using MobileNet SSD on video stream is a very crucial topic of surveillance systems in field applications. To easily accessible product, the project constructed as low cost project. In this project, several methods are presented.

We implemented different detecting methods. Algorithms works well for different detection purposes. The results are good for starting. MobileNet SSD which is based on CNN model gives the best result for our project.

REFERENCES

http://cocodataset.org/#explore

https://www.opencv-srf.com/p/introduction.html

https://ebenezertechs.com/mobilenet-ssd-using-opencv-3-4-1-deep-learning-module-python/

Research Papers:

MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications Andrew G. Howard Menglong Zhu Bo Chen Dmitry Kalenichenko Weijun Wang Tobias Weyand Marco Andreetto Hartwig Adam

Study on Object Detection using Open CV - Python
Bhumika Gupta, PhD Assistant Professor, C.S.E.D G.B.P.E.C, Pauri Uttarakhand, India Ashish Chaube B.Tech IV Year G.B.P.E.C, Pauri Uttarakhand, India Ashish Negi B.Tech IV Year G.B.P.E.C, Pauri Uttarakhand, India Umang Goel

Deep Learning in Neural Networks: An Overview Technical Report
Jurgen Schmidhuber ¨ The Swiss AI Lab IDSIA Istituto Dalle Molle di Studi sull'Intelligenza Artificiale University of Lugano & SUPSI Galleria 2, 6928 Manno-Lugano Switzerland