**Honey Bees classification using Deep learning**

**Abstract**: This project includes implementation of deep learning based architectures for classification of four types of honeybees classification of bees was highly required to understand the characterstics of them which can be used to determine and detect the health of the Bee hive’s,honeybess are the most important Pollinators in our ecosystem.By this we can boost the process of Pollination which will result in high quality food products for us .Popular CNN based architectures were evaluated on the basis of classification accuracy and prediction speed.Based on the results a simple yet effective Convolutional Neural Network (CNN) was proposed with some high accuracy and some decent prediction speed.

**1.Introduction**.

Every third bite of food relies on pollination by bees. The most important thing that bees do is pollinate. Pollination is needed for plants to reproduce, and so many plants depend on bees or other insects as pollinators. At the same time, this past winter honeybee hive losses have exceeded 60% in some states. How can we address this issue? How can we better understand our bees? Can CNN make an impact on our problem? Will it able to improve pollination process? And most importantly, how can we save them before it's too late?While many indications of hive strength and health are visible on the inside of the hive, frequent check-ups on the hive are time-consuming and disruptive to the bees' workflow and hive in general. By investigating the bees that leave the hive, we can gain a more complete understanding of the hive itself. For example, an unhealthy hive infected with varroa mites will have bees with deformed wings or mites on their backs. These characteristics can be observed without opening the hive.To protect against robber bees, we could track the ratio of pollen-carrying bees vs those without. A large influx of bees without pollen may be an indication of robber bees. This dataset aims to provide basic visual data to train machine learning models to classify bees in these categories, paving the way for more intelligent hive monitoring or beekeeping in general. We began by studying the dataset at hand. Data preprocessing was done to make the data ready to be used. First, data was augmented to equalize class imbalance and standardization was applied to it. Models were trained and tested on this preprocessed data. Throughout this project, our main focus was to increase complexity step by step and figure out what works the best for this dataset.

**2.Methodology.**

When computer vision started to take shape as a field in the 1960s, its aim was to try and mimic human vision systems and ask computers to tell us what they see, automating the process of image analysis. This kind of technology is the precursor to artificially intelligent image recognition. Before, any kind of image analysis had to be done manually, from x-rays to MRIs to hi-res space photography.

Just like animals, computers “see” the world differently from us humans: basically, they count the number of pixels, try to discern borders between objects by measuring shades of color, and estimate spatial relations between objects.

As computer vision evolved, algorithms started to be programmed to solve individual challenges, and they become better at doing the job the more they repeat the task.

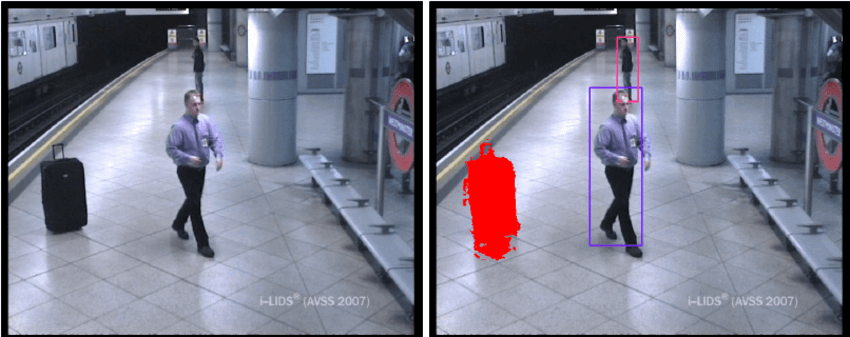


Fig 1.a) CCTV cameras detecting objects

Fast forward to 2010 (and beyond), we have seen an acceleration in improved deep learning techniques and technology. With deep learning, we’re now able to program supercomputers to train themselves, self-improve over time and provide portions of these capabilities to businesses as online applications, like cloud-based apps.In order for these machines to learn, they need to be fed data.

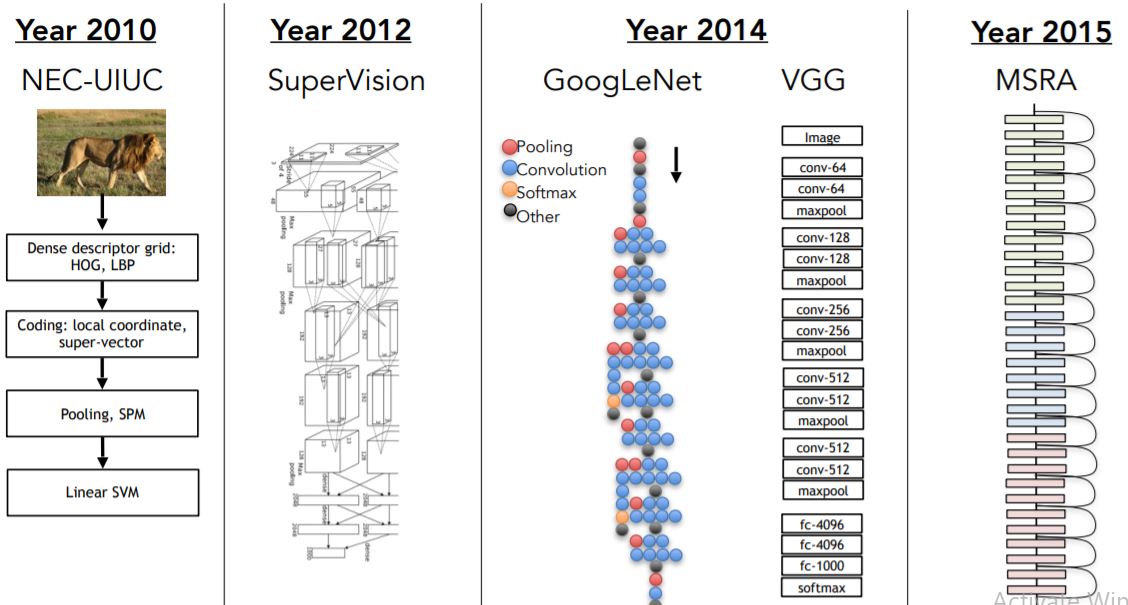


Fig 1.b) Evolution of Computer Vision

What distinguished computer vision from the prevalent field of digital image processing at that time was a desire to extract three-dimensional structure from images with the goal of achieving full scene understanding. Studies in the 1970s formed the early foundations for many of the computer vision algorithms that exist today, including extraction of edges from images, labeling of lines, non-polyhedral and polyhedral modeling, representation of objects as interconnections of smaller structures, optical flow, and motion estimation.

ImageNet is a dataset of over 15 million labeled high-resolution images belonging to roughly 22,000 categories. The images were collected from the web and labeled by human labelers using Amazon’s Mechanical Turk crowd-sourcing tool. Starting in 2010, as part of the Pascal Visual Object Challenge, an annual competition called the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been held. ILSVRC uses a subset of ImageNet with roughly 1000 images in each of 1000 categories. In all, there are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images. ILSVRC-2010 is the only version of ILSVRC for which the test set labels are available, so this is the version on which we performed most of our experiments. Since we also entered our model in the ILSVRC-2012 competition, in Section 6 we report our results on this version of the dataset as well, for which test set labels are unavailable. On ImageNet, it is customary to report two error rates: top-1 and top-5, where the top-5 error rate is the fraction of test images for which the correct label is not among the five labels considered most probable by the model. ImageNet consists of variable-resolution images, while our system requires a constant input dimensionality. Therefore, we down-sampled the images to a fixed resolution of 256 × 256. Given a rectangular image, we first rescaled the image such that the shorter side was of length 256, and then cropped out the central 256×256 patch from the resulting image. We did not pre-process the images in any other way, except for subtracting the mean activity over the training set from each pixel. So we trained our network on the (centered) raw RGB values of the pixels.

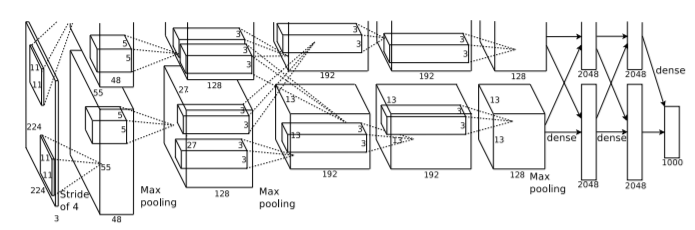


Fig 1.c) Architecture of Imagenet

AlexNet was much larger than previous CNNs used for computer vision tasks ( e.g. Yann LeCun’s LeNet paper in 1998). It has 60 million parameters and 650,000 neurons and took five to six days to train on two GTX 580 3GB GPUs. Today there are much more complex CNNs that can run on faster GPUs very efficiently even on very large datasets. But back in 2012, this was huge!

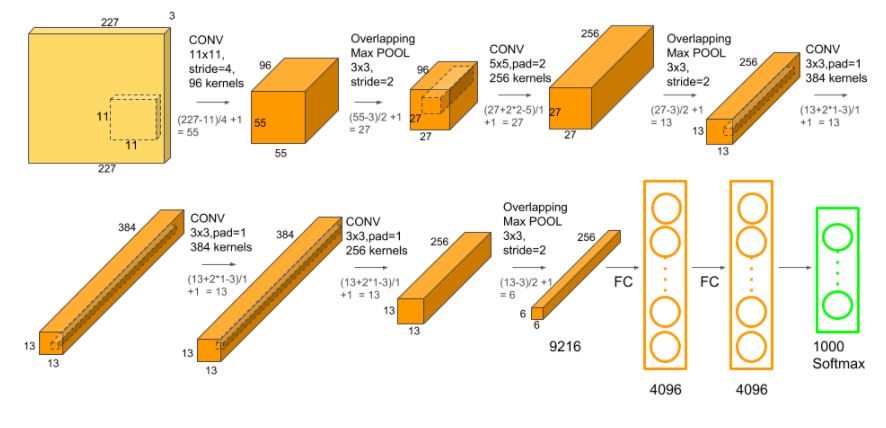


Fig 1.d) Architecture of Alexnet

Multiple Convolutional Kernels (a.k.a filters) extract interesting features in an image. In a single convolutional layer, there are usually many kernels of the same size. For example, the first Conv Layer of AlexNet contains 96 kernels of size 11x11x3. Note the width and height of the kernel are usually the same and the depth is the same as the number of channels.

The first two Convolutional layers are followed by the Overlapping Max Pooling layers that we describe next. The third, fourth and fifth convolutional layers are connected directly. The fifth convolutional layer is followed by an Overlapping Max Pooling layer, the output of which goes into a series of two fully connected layers. The second fully connected layer feeds into a softmax classifier with 1000 class labels.

ReLU nonlinearity is applied after all the convolution and fully connected layers. The ReLU nonlinearity of the first and second convolution layers are followed by a local normalization step before doing pooling. But researchers later didn’t find normalization very useful. So we will not go in detail over that.

The next decade saw studies based on more rigorous mathematical analysis and quantitative aspects of computer vision. These include the concept of scale-space, the inference of shape from various cues such as shading, texture and focus, and contour models known as snakes. Researchers also realized that many of these mathematical concepts could be treated within the same optimization framework as regularization and Markov random fields. By the 1990s, some of the previous research topics became more active than the others. Research in projective 3-D reconstructions led to better understanding of camera calibration. With the advent of optimization methods for camera calibration, it was realized that a lot of the ideas were already explored in bundle adjustment theory from the field of photogrammetry. This led to methods for sparse 3-D reconstructions of scenes from multiple images. Progress was made on the dense stereo correspondence problem and further multi-view stereo techniques. At the same time, variations of graph cut were used to solve image segmentation. This decade also marked the first time statistical learning techniques were used in practice to recognize faces in images. Toward the end of the 1990s, a significant change came about with the increased interaction between the fields of computer graphics and computer vision. This included image-based rendering, image morphing, view interpolation, panoramic image stitching and early light-field rendering.

Starting with LeNet-5 , convolutional neural networks (CNN) have typically had a standard structure – stacked convolutional layers (optionally followed by con

Recent work has seen the resurgence of feature-based methods, used in conjunction with machine learning techniques and complex optimization frameworks. The advancement of Deep Learning techniques has brought further life to the field of computer vision. The accuracy of deep learning algorithms on several benchmark computer vision data sets for tasks ranging from classification, segmentation and optical flow has surpassed prior methods.

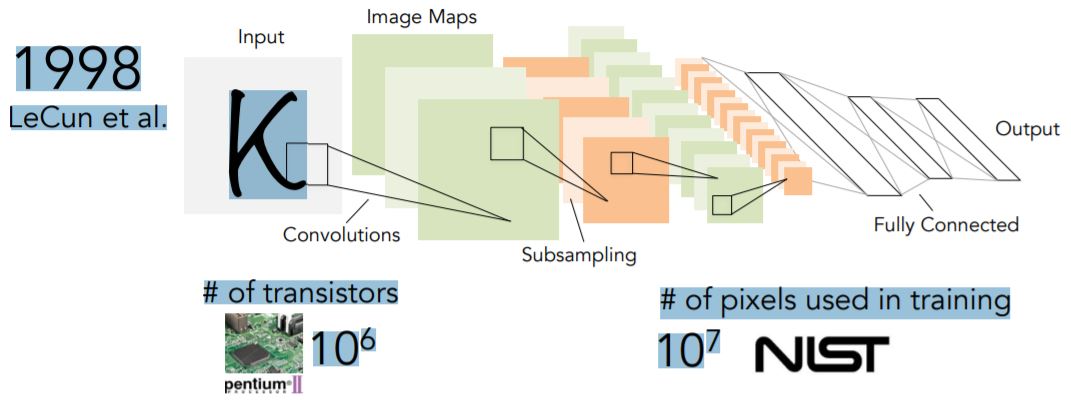


Fig 1.c) Yann Le cun research paper’s CNN network

Artificial intelligence has already been seamlessly integrated into many aspects of our daily lives. AI has found tremendous success in many areas of research in recent years. Game-playing systems like AlphaGo have used reinforcement learning to teach themselves new strategies.  Hearing aides use deep learning algorithms to filter out ambient noise.  These technologies even power the natural language processing and translation, object recognition, and pattern matching systems that we take for granted on Google, Amazon, iTunes, and similar services.

This trend shows no signs of slowing down – there are many small, repetitive tasks that we can automate to free up our time.  Although we have made incredible strides in the field of artificial intelligence, we still need to be realistic about its applications for computer vision – it will be a long time before computers are able to interpret images as well as humans can.

**3.Simulations**

**3.1 Dataset Information**

The Dataset used here is Annotated HoneyBee Image Dataset from Kaggle which consisted of 4 categories of Bee images of dimension (72x72x3),3 signifying the RGB channel of colored images .The dataset consisted of 5173 images in which 4143 was used for training the model,500 for validation and 500 for test.Below are some sample images from our dataset.

Fig.3.1 a) Sample Images from Dataset

**3.2 Data Preprocessing**

In preprocessing we have done data augmentation. Images from low frequency classes were picked up and random rotation and brightness variation was performed. Eventually, data was normalized for all classes . Then, standardization was performed on the data to normalize its mean and make it unit variance.Also we need to convert the images into array so that our deep learning algorithms can be applied and further can be fed to our ANN for forward and backward propogation. During training, the machine adjusts its internal parameters to project each feature tensor close to its target.

After training, the machine can be used to predict the target for previously unseen feature tensors.What this study focuses on is the requirement that feature tensors must be of the same size.In other words, the same number of features must be present for each sample.Hence we need to resize every image present in our dataset



array ([[[130., 165., 142.],

[130., 165., 142.],

[130., 165., 142.

[ 56., 82., 59.],

[ 56., 82., 59.],

[ 56., 82., 59.]],

[[130., 165., 142.],)

Fig 3.2 a) The above image is converted into an array to feed to a neural network

**3.3 Model Implementation**

In this section, we will discuss the CNN model implemented on our dataset for classification in detail.

Here we can also see if pixels have any sequential relation which can be exploited to boost the performance

**3.3.1 Proposed Convolutional Neural Network**

A**Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

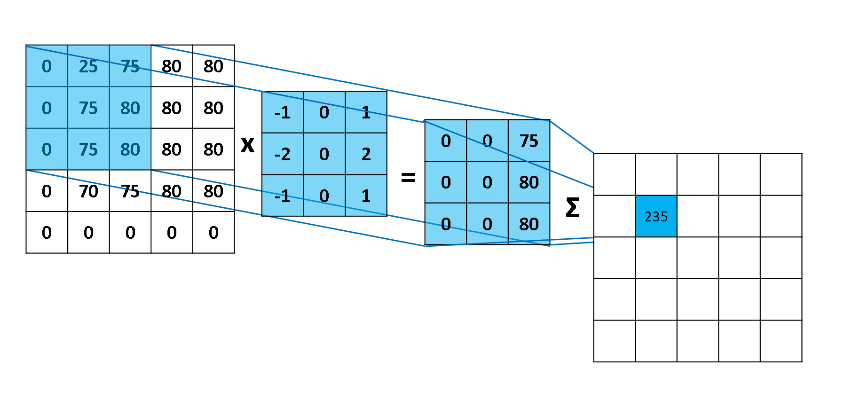


Fig 3.3.1.a) Convolution Operation in CNN

In Convolutional Neural Networks, Filters detect spatial patterns such as edges in an image by detecting the changes in intensity values of the image.

In terms of an image, a high-frequency image is the one where the intensity of the pixels changes by a large amount, whereas a low-frequency image is the one where the intensity is almost uniform. Usually, an image has both high and low frequency components. The high-frequency components correspond to the edges of an object because at the edges the rate of change of intensity of pixel values is high .**High pass filters** are used to enhance the high-frequency parts of an image.

Padding is a term relevant to convolutional neural networks as it refers to the amount of pixels added to an image when it is being processed by the kernel of a CNN.

An activation function is a very important feature of an artificial neural network , they basically decide whether the neuron should be activated or not. In artificial neural networks, the activation function defines the output of that node given an input or set of inputs.

There are two types of Pooling: Max Pooling and Average Pooling. **Max Pooling** returns the **maximum value from** the portion of the image covered by the Kernel. On the other, **Average**returns the **average of all the values**from the portion of the image covered by the Kernel.

We implemented a convolution neural network architecture that has two stacks made up of three convolution layers followed by max pooling layer after each convolutional layer and two fully connected layers. We used batch normalization after every convolution layer to avoid exploding and vanishing of gradients while back propagating across different layers ReLU activation function was used as a nonlinearity after each convolution layer which set negative input to zero units. The convolution layers have 16, 32,64 convolution filters with kernel size = 2 and learning rate=0.0001. Its output is given to a fully connected hidden layer consisting of 500 neurons. Final output layer has 4 neurons with softmax as activation function to calculate the probability corresponding to 4 classes of our dataset.

**4.Result’s**

The test accuracy achieved here is 96.23% for our model with 20 epochs

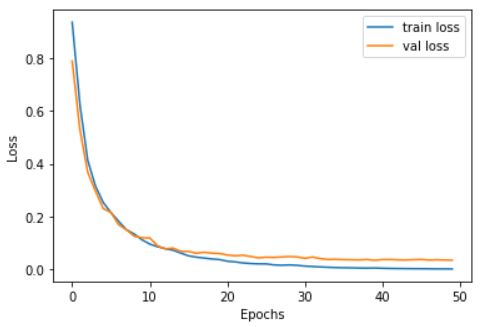
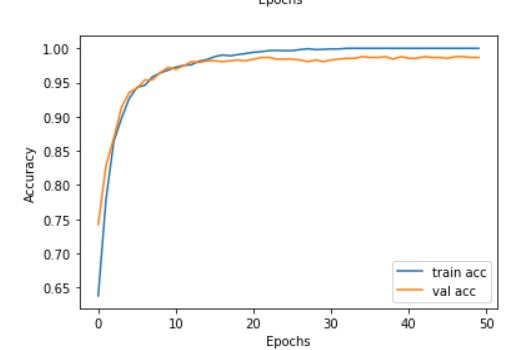
 

Fig 4 a) Accuracy and Loss Curves for 50 epochs

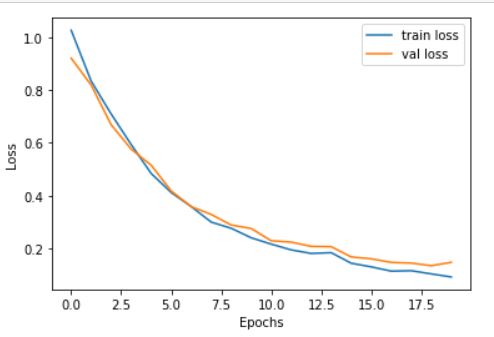
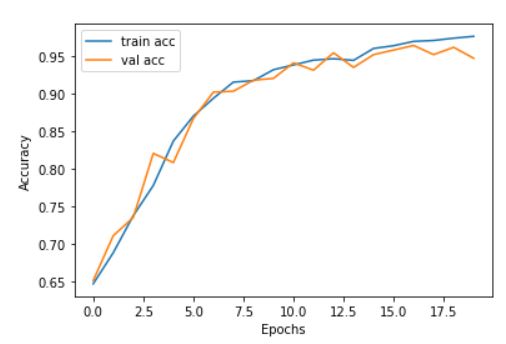
** **

Fig 4 b) Accuracy and Loss Curves for 20 epochs

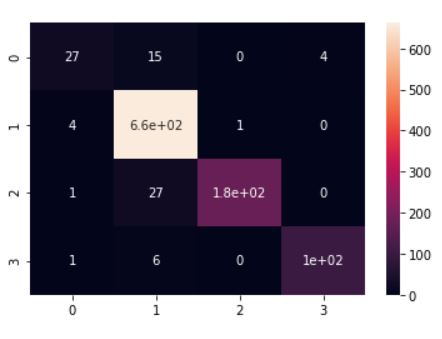


Fig 4 c) Confusion Matrix

**4.1 Analysis of Result**

For training of our model first we took 50 epochs with batch\_size of 128 the training accuracy that we got was around 100 % with loss of 0.0027 and validation accuracy of 98.66 % with validation loss of 0.0359 here we observed that after a particular number of epoch there was no increase in the accuracy score and there was no decrease in our loss function.Hence we did early stopping of our model .

Training our model with 50 epochs will make no sense if there is no change in our parameters hence we decided to decrease our epochs to 20.After that we got training accuracy of 97.21% and validation accuracy of 94.95% with loss of 0.0377 and 0.0978 for training and validation respectively.

And test accuracy of 94.26% which seems that our model is performing well and is a generalized one.

Also by analysing confusion matrix we have few misclassified classes in our result.

**5.Conclusions**

In this project we have implemented Convolutional Neural network and validated our model with the

accuracy metric.Also we observed we can take few epochs to achieve a good result.Here we can now classify our images into different categories according to its species or body type from that we can

study the characterstics of that particular .

By which we can improve the health of the bee hives and keep the bees out which damage or harm the bee hives.Here we can also improve pollination and give importance to the bees which are contributing to improve the health of the hive with the help of this we can achieve a good quality of fruits,vegetables and other food items which can be very much beneficial for our community.

**6.References**

a) X. Wang, C. Chen, Y. Cheng, et al, Zero-shot image classification based on deep feature extraction.

United Kingdom: IEEE Transactions on Cognitive & Developmental Systems, 10(2), 1–1 (2018).

b) Shima Y. Image augmentation for object image classification based on combination of pre-trained CNN and SVM. International Conference on Informatics, Electronics and Vision & 2017, International sSymposium in Computational Medical and Health Technology. 2018:1–6.

c) M.Z. Afzal, A. Kölsch, S. Ahmed, et al., Cutting the error by half: investigation of very deep CNN and advanced training strategies for document image classification

(Iapr international conference on document analysis and recognition. IEEE computer Society, Kyoto, 2017), pp. 883–888.

d) Z.Yan, V.Jagadeesh, D.Decoste, et al., HD-CNN: hierarchical deep convolutional neural network for image classification.

Eprint Arxiv 4321-4329 (2014).

e) S. Roychowdhury, J. Ren, Non-deep CNN for multi-modal image classification and feature learning: an azure-based model

(IEEE international conference on big data. IEEE, Washington, D.C., 2017), pp. 2893–2812.

f) Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang,

Baoli Li, and Shimin Hu. Traffic-sign detection and classification in the wild.

In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

g) Pierre Sermanet, David Eigen, Xiang Zhang, Michael

Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization

and detection using convolutional networks, 2013.