**Image Analysis Using Deep Learning**

**Synopsis**

**Submitted by**

**Aditya R. Patgaonkar**

**149107154**

**Under the Supervision of**

**Dr. Himanshu Chaudhary**

**(Professor, Electronics & Communications)**



B.Tech / Electronics & Communications

MANIPAL UNIVERSITY JAIPUR  
JAIPUR-303007  
RAJASTHAN, INDIA

January/2018

Project Summary

|  |  |
| --- | --- |
| *Organization Details* | |
| Company Name | Mercedes Benz Research & Development India |
| Full postal address with pin code | Plot No. 9 & 10, Whitefield Palms, EPIP Zone, Phase 1, Whitefield Road, Near Satya Sai Baba Hospital, Bengaluru, Karnataka 560066 |
| Website address | http://mbrdi.co.in |
|  |  |
| *Supervisor Details* | |
| Supervisor Name | Dr. Hisham Cholakkal, Dr. Sanath Narayan |
| Designation | Senior Technical Lead (Deep Learning Expert) |
| Phone Number (M) | +91 8061494029 |
| Email address | [hisham.cholakkal@daimler.com](mailto:hisham.cholakkal@daimler.com), [sanath.narayan@daimler.com](mailto:sanath.narayan@daimler.com) |
|  |  |
| *Project Details* | |
| Project Title | Image Analysis Using Deep Learning |
| Project Duration *(in weeks)* | 24 weeks |
| Date of project commencement | 2nd January |
| Supervisor Sign with Seal |  |

**1. Objective of work**

Research & Development for Computer Vision using Deep Learning for Car Electronics Infotainment Systems

**2. Motivation**

Improving image segmentation techniques using state of the art deep learning techniques.

**3. Functional partitioning of project**

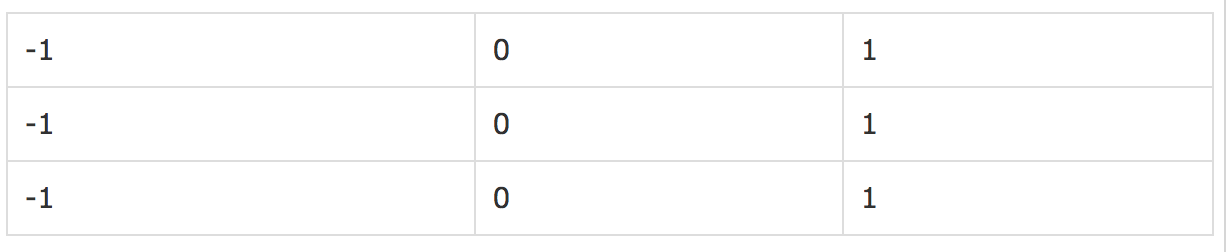
* 1. **Image filtering**:[7]
* Prewitt operator:

Based on derivative masks. This mask uses fixed coefficients.

For horizontal and vertical edge detection

Vertical edge detector filter:

Figure 1



Horizontal edge detector:

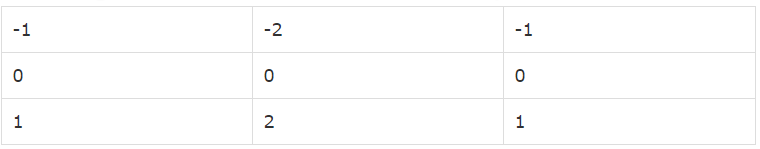
Figure 2



* Sobel operator:

The coefficients of masks are not fixed and they can be adjusted according to our requirement unless they do not violate any property of derivative masks.

Figure 3

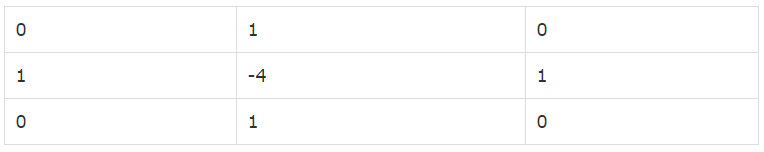


* Laplacian operator:

*Positive Laplacian Operato*r:

In Positive Laplacian we have standard mask in which center element of the mask should be negative and corner elements of mask should be zero.

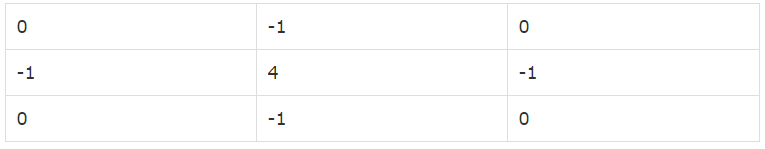
Figure 4



## *Negative Laplacian Operator*:

In negative Laplacian operator we also have a standard mask, in which center element should be positive. All the elements in the corner should be zero and rest of all the elements in the mask should be -1.

Figure 5



* Gabor filter:

Linear filter for texture (representation & discrimination) analysis. Texture analysis analyses if there are any specific frequency content in the image in specific directions in a localized region around the point or region of analysis.

* Gaussian blur:

Similar to a low pass filter. This filter enables blurring of an image via a Gaussian function.

Reduces image noise and detail.

It is used as a pre-processing stage in computer vision algorithms to enhance image structures at different scales.

* Image features:

Features may include: Points, edges, objects, motion in image sequences, shapes, curves, boundaries between different image regions and properties of such regions. Some of the common types are described below.

*Edges*: Boundary between 2 image regions. Arbitrary shape. May include junctions. Sets of points in the image with a strong gradient magnitude. Algorithms chain high gradient points to form a more complete description of an edge.

*Corners / Interest points*: Rapid changes in directions (corners) in edges. Algorithms are developed to eliminate dependency on edge detection for points of interest. Corner detection involves looking for high levels of curvature in the image gradient. However “non-corners” i.e. bright spots on a dark background may be detected. That is why, the term region of interests comes into consideration.

* Basic machine learning algorithm for image classification:

*K-Nearest Neighbor (KNN):*

Widely used in classification problems. It can also be used in regression problems.

Classification is widely implemented in the industry since most problems involve taking a decision.

* + KNN fares well in terms of ease to interpret output, calculation time and predictive power.
  + KNN is a non-parametric lazy learning algorithm.
  + KNN is never used on images since it has a terrible performance at test time (space inefficient)

Distance metrics on a level of whole images can be very unintuitive.

* 1. **Deep learning:** [1]
* Artificial intelligence:

The state of the art artificial intelligence based products are based on deep networks and learning

* Why is it useful?

Solves previously unsolvable problems like human pose estimation, lip reading, image caption generation, human computer interaction.

* Based on: A data driven paradigm. Why is this paradigm useful? If we consider an image classification task, this paradigm enables the classifier to classify images correctly while being –
  + Invariant to illumination.
  + Invariant to viewpoint.
  + Deals with occlusion.
  + Invariant to deformation.
  + Deals with background clutter.
  + Deals with intra-class variation.

For accurate classification tasks, there is no way to hard-code the algorithms, which is why the data driven paradigm is so important for deep learning.

An image classification task is implemented in the following way using this paradigm:

* + Collect a dataset of images and labels.
  + Use machine learning to train an image classifier.
  + Evaluate the classifier on a withheld set of test images.

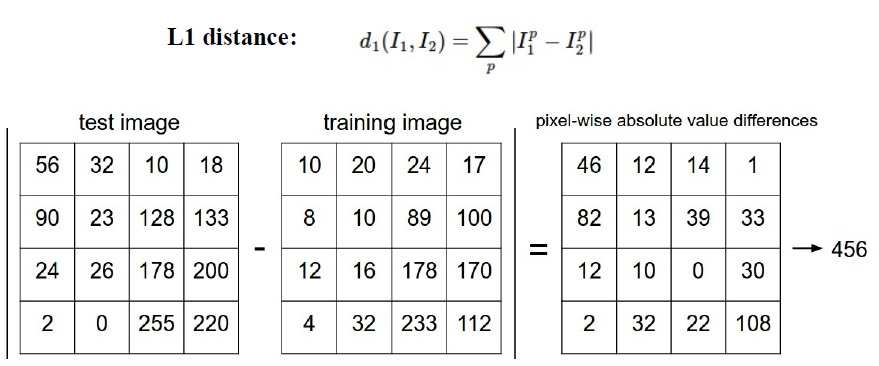
* Basic algorithms:

*Nearest neighbor classifier****:***

Remember all the training images and their labels and predict the label of the most similar training image.

Use the L1 distance metric to calculate the difference between two images and choose the most similar image as the prediction.

Figure 6



*Hyper parameters*: Mostly problem dependent. The validation data is used to set the hyper parameters.

The nearest neighbor classifier’s classification speed depends on the data size i.e. linearly, which is not good. In practice, test time performance is very important.

*Parametric approach*:

* Linear classifier:

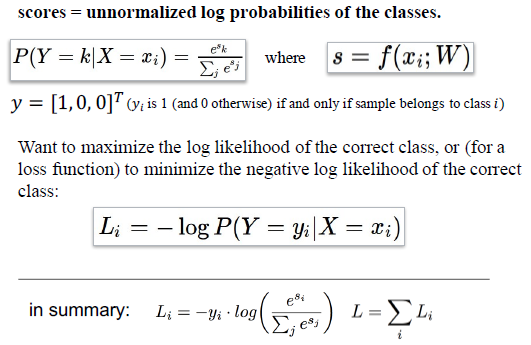
Image 🡪 f (*x, W) = Wx + b* 🡪 10 numbers indicating class scores

f (*x, W) = Wx + b:* Linear scoring function.

Going forward we will need a loss function & optimization.

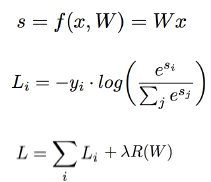
* Softmax classifier (Multinomial logistic regression):

Figure 7



*Weight regularization:*

Figure 8



λ: The regularization strength.

*Optimization & Gradient Descent*:

Figure 9

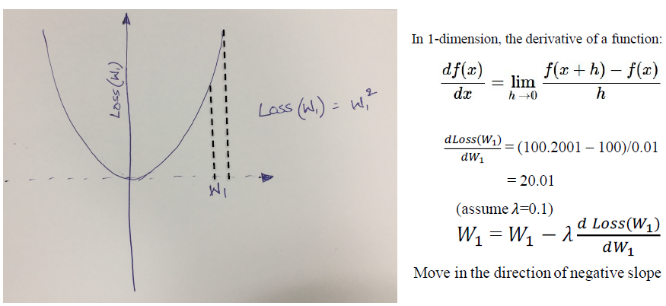
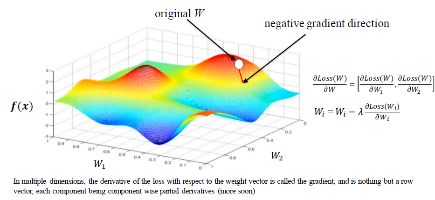


Figure 10



* Recurrent Neural Networks (RNNs):
* Sequential neural network that operates using sequential information.
* Inputs & outputs are dependent on each other.
* The “memory” which captures information is used.
* The network shares same parameters along all the steps.
* RNNs & its extensions are used for applications like natural language processing, language modelling & text generation, machine translation, speech recognition, and generate image descriptions.
* LSTM: Long short term memory network is an example of an RNN extension.
  1. **Deep learning for computer vision:**
* Convolution NNs (CNN/ConvNet):[6]

Convolutional Neural Networks are very similar to ordinary Neural Networks from the previous chapter: they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The difference come because ConvNet architectures make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture. These then make the forward function more efficient to implement and vastly reduce the amount of parameters in the network.

Figure 11



*Activation function:*

ReLU. The Rectified Linear Unit has become very popular in the last few years. It computes the function f(x) = max (0, x). In other words, the activation is simply thresholded at zero.

Figure 12



* Object detection techniques**:**
* RCNN [1]: Region proposals (RoI: Region of Interest) + Convolutional neural networks.

Image 🡪 RoIs (Selective Search) 🡪 CNNs 🡪 Classification (SVMs) 🡪 Bounding Boxes (SVMs)

Prediction of region of interests (RoIs) in an image (objects) and classifying them.

* Fast RCNN [2]: Extension of RCNN

Image 🡪 CNN 🡪 RoIs (Selective Search) 🡪 RoI Pooling 🡪 Classification (Softmax) & Bounding Boxes (Regression)

* Faster RCNN [3]: Extension of Fast RCNN. Adds a Region Proposal Network (RPN)

Image 🡪 CNN 🡪 RPN 🡪 RoI Pooling 🡪 Classification (Softmax) & Bounding Boxes (Regression)

* R-FCN [4]: Region Proposals + Fully Convolutional Network

Image 🡪 CNN 🡪 RPN + Specialized Conv Layers 🡪 RoI Pooling 🡪 Classification (Softmax) & Bounding Boxes (Regression)

* Mask RCNN [5]: Semantic segmentation i.e. pixel level segmentation.

Image 🡪 CNN 🡪 RPN 🡪 RoI Align 🡪 Masks + Classification (Softmax) & Bounding Boxes (Regression)

**5. Methodology**

Read & understand research papers based on the concepts relevant for the project. Discuss them with the team to plan and decide the implementation. Implement the algorithms using the decided framework.

**6. Tools required**

*Linux*: Ubuntu 14.04

*Torch*: Scientific computing framework with wide support for machine learning algorithms that puts

GPUs first. It is easy to use and efficient, thanks to an easy and fast scripting language, LuaJIT, and an

underlying C/CUDA implementation.

*Pytorch*: Python based package for Tensors and dynamic neural networks in Python

with strong GPU acceleration.

*Nvidia GPUs (CUDA)*: Framework to enable Nvidia GPUs for computational tasks etc.

**7. Work schedule *(month wise)***

## (a) Jan 2018

Monday-Friday, 11 hours (between 7am - 7pm)

(b) Feb 2018

Monday-Friday, 11 hours (between 7am - 7pm)

(c) Mar 2018

Monday-Friday, 11 hours (between 7am - 7pm)

(d) Apr 2018

Monday-Friday, 11 hours (between 7am - 7pm)

(e) May 2018

Monday-Friday, 11 hours (between 7am - 7pm)

(f) June 2018

Monday-Friday, 11 hours (between 7am - 7pm)

**8. References**

*Journal / Conference Papers*

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014

[2] R. Girshick. Fast R-CNN. In ICCV, 2015.

[3] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.

[4] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object detection via region-based fully convolutional networks. In NIPS, 2016.

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. Mask R-CNN

*Web*

[1] Deep Learning (for Computer Vision - CS 763) Module - Spring 2017.

https://github.com/cs763-dl/2017Spring

[2] Difference between object-detection, semantic-segmentation & localization.

https://cs.stackexchange.com

[3] Brief history of CNNs for object detection. http://cs231n.github.io/

[4] Deep Learning for Object Detection. https://towardsdatascience.com/

[5] Deep Learning. Computer Science Department, Stanford University. http://ufldl.stanford.edu/

[6] CS231n: Convolutional Neural Networks for Visual Recognition. http://cs231n.github.io/

[7] Digital Image Processing, Python OOPs, Lua: https://www.tutorialspoint.com/