Object Classification Using Neural Network and Various Feature Extractors

ZHAO, Xiaotian
趙　笑添

Advisor: Kuticsne Matz, Andrea

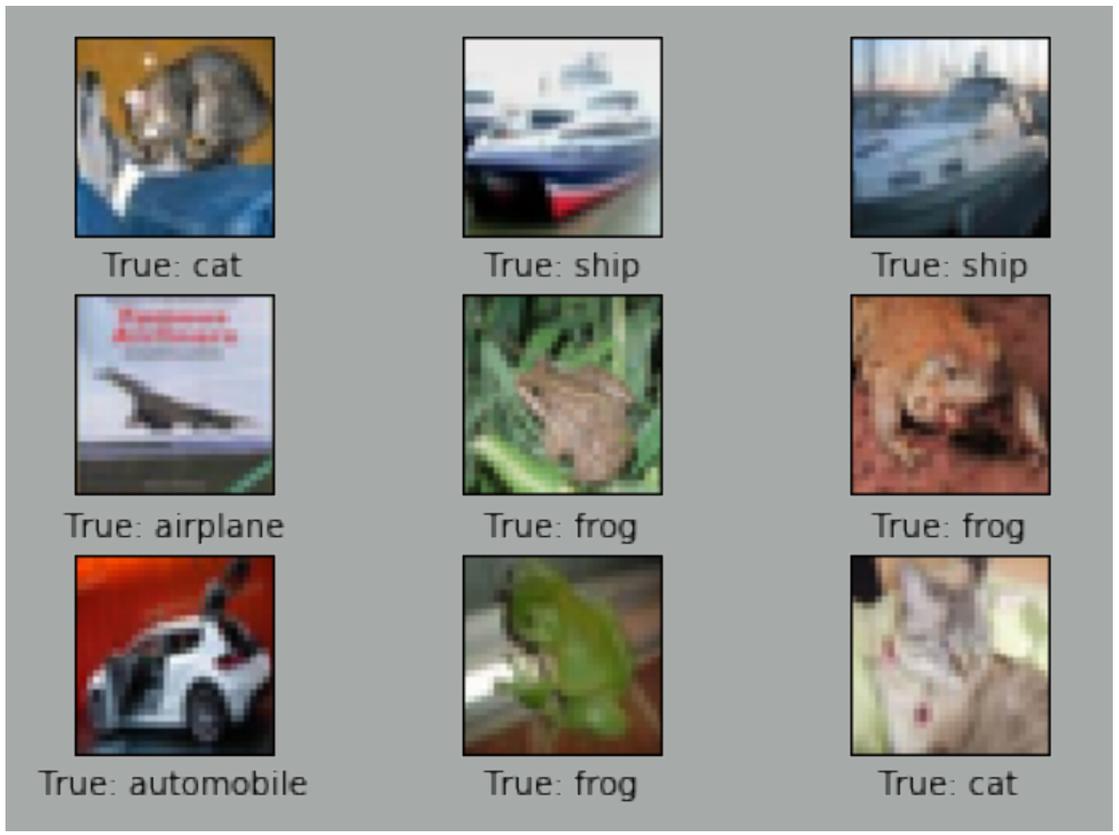
Outline of this presentation

1. Introduction
2. Related Works
3. Our Model
4. Implementation
5. Experiments

Introduction

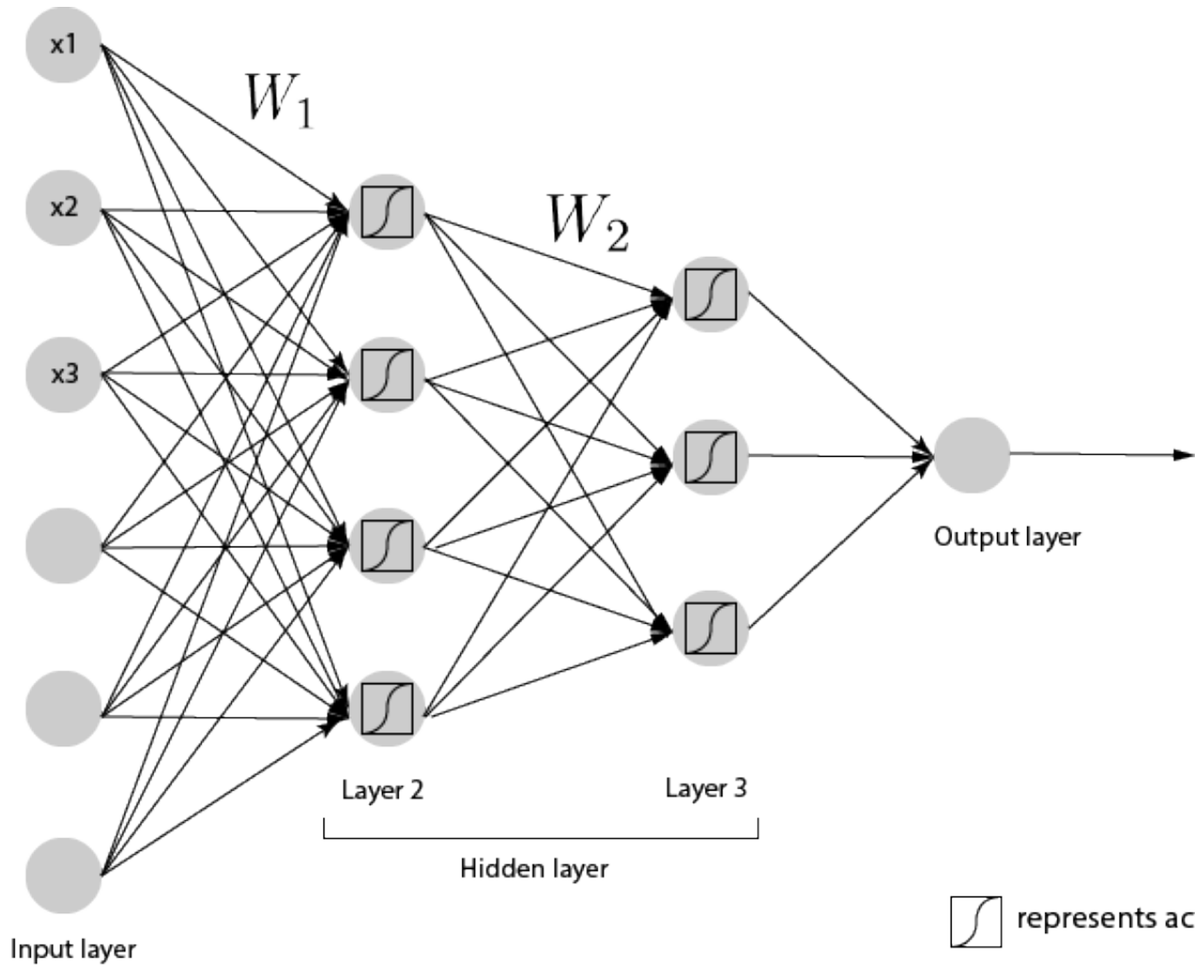
Object Classification

Classify Objects from various categories. Each image will contain a single target object.



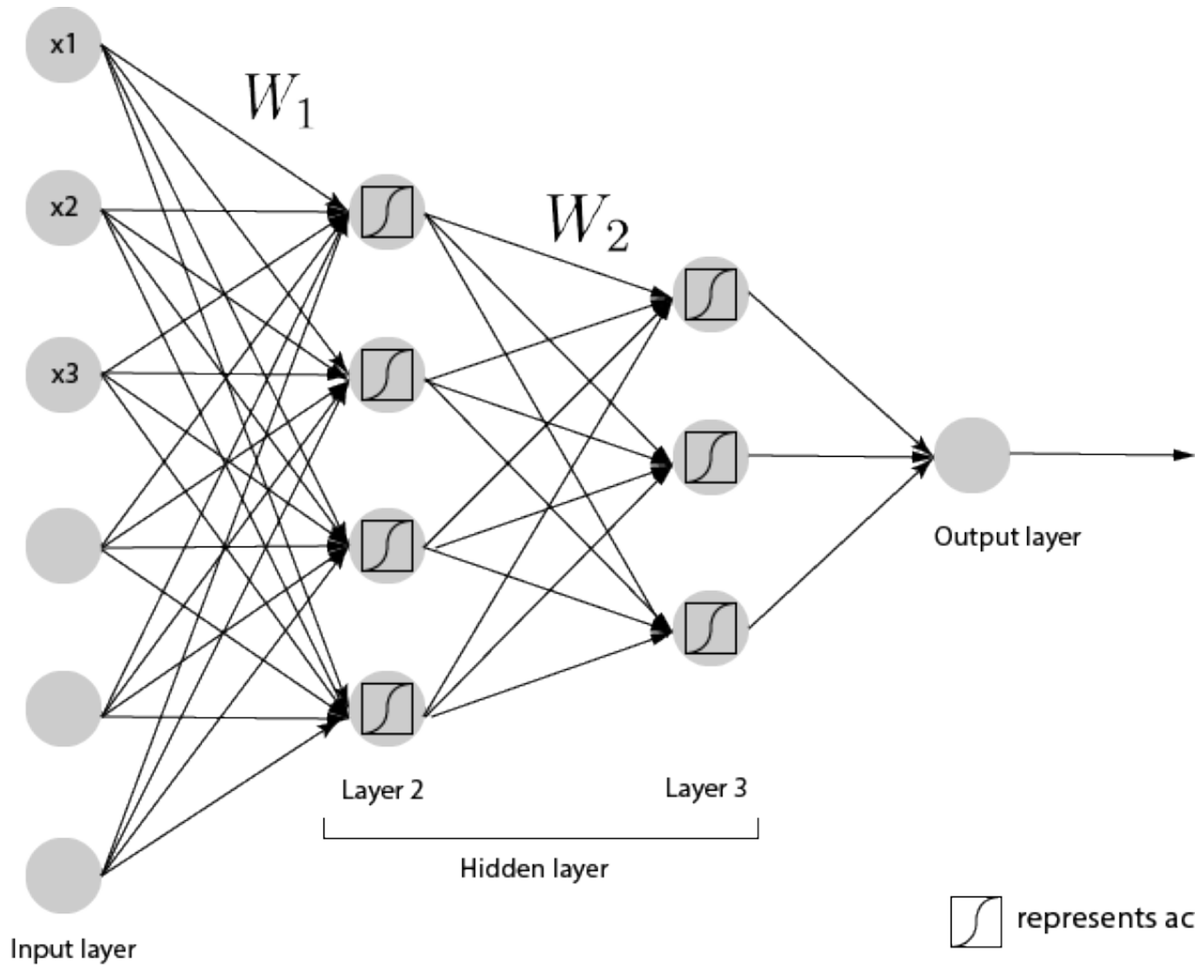
Problem with existing methods

- Almost all the features of an image are extracted with automated process
- Region boundaries (shape) of objects are difficult to determine without explicit handcrafted feature extractor.
- Different Object, Different Focused Feature. However automated process can not handle that.



Motivation

- Fairly accurate region boundaries (shape) of objects can be extracted with handcrafted feature extractors.
- Neural network with such feature extractor should perform better than original image.



Related Works

1. Feature Extractors
2. Artificial Neural Network based Classification

Feature Extractors

Dominant Color Descriptor

Color Structure Descriptor

Color Layout Descriptor

Region-based Shape Descriptor

Homogeneous Texture Descriptor

Contour-based Shape Descriptor

Texture Browsing Descriptor

Canny Edge Descriptor

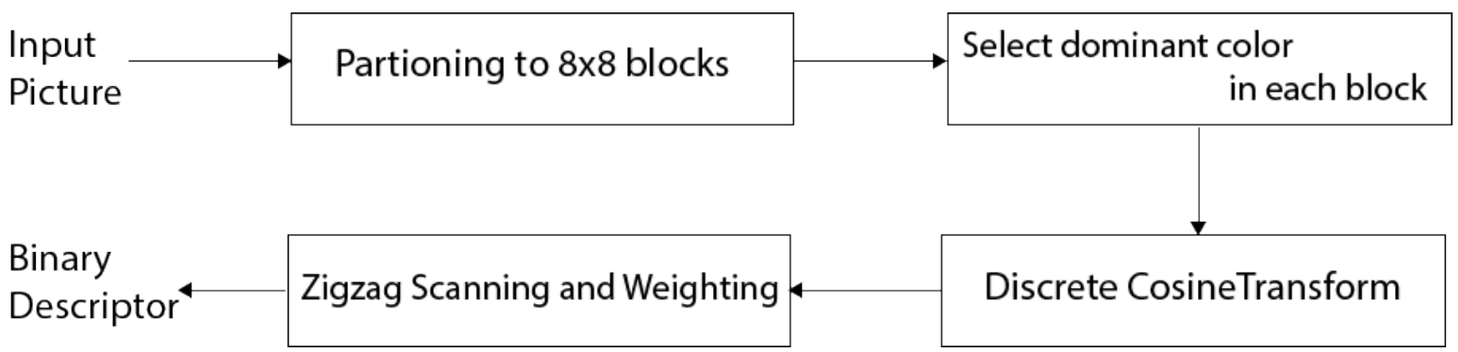
Anisotropic Diffusion

Ultrametric Contour Map

Color Layout Descriptor

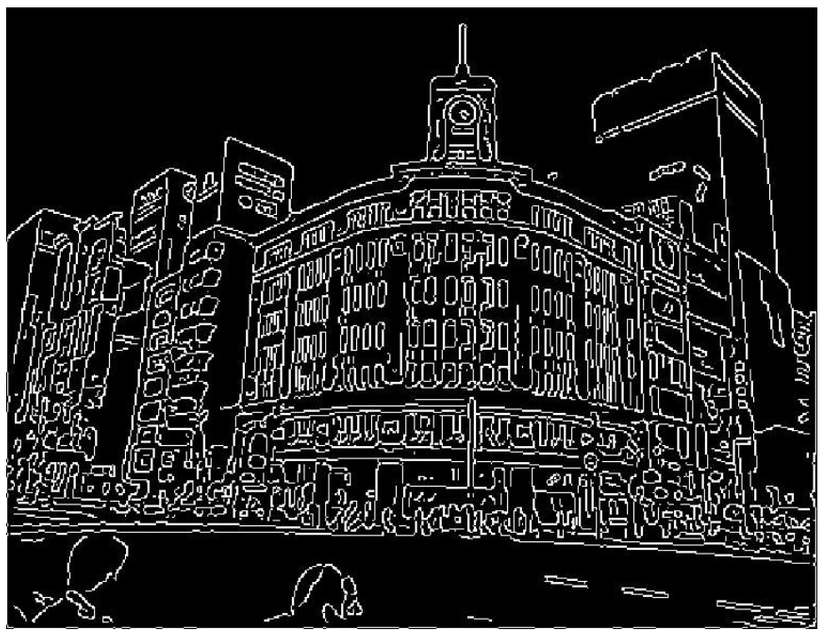
- Capable of representing spacial distribution of color with small data size.
- Applied in similar images and video retrieval
- Can be used as one of the image abstraction method

https://upload.wikimedia.org/wikipedia/commons/thumb/f/f2/Representative_color_selection_2.jpg/300px-Representative_color_selection_2.jpg



Canny Edge Extractor

- Highly optimized edge extractor that uses differential, orientation and gaussian smoothing.
- Capable of extracting edge, but not contour or shape of the Objects



Anisotropic Diffusion

- For extracting region boundaries
- Blur complicated texture of objects but maintain the important region boundaries
- Computationally heavy

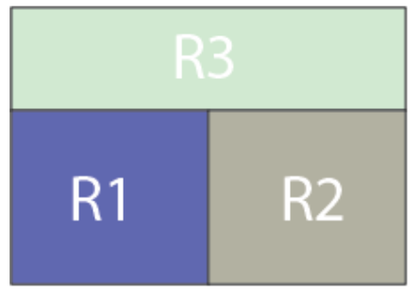
Ultrametric Contour Maps

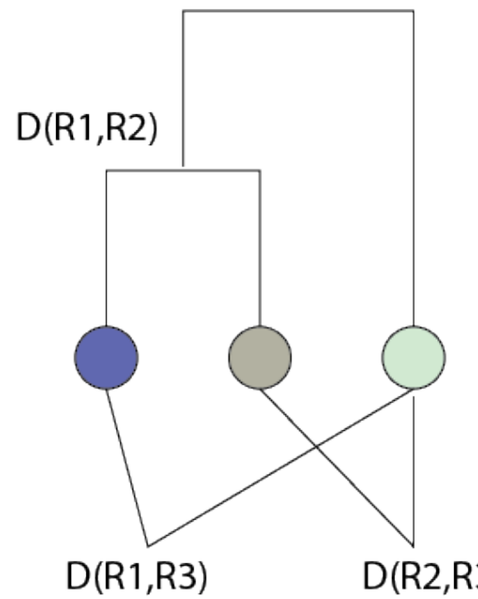
- Used to extract region boundaries and segmentation tasks

- Construct a union algorithm like graph based on the color distance between each regions

- Color that are close in distance will be merged to one region

- Computationally light





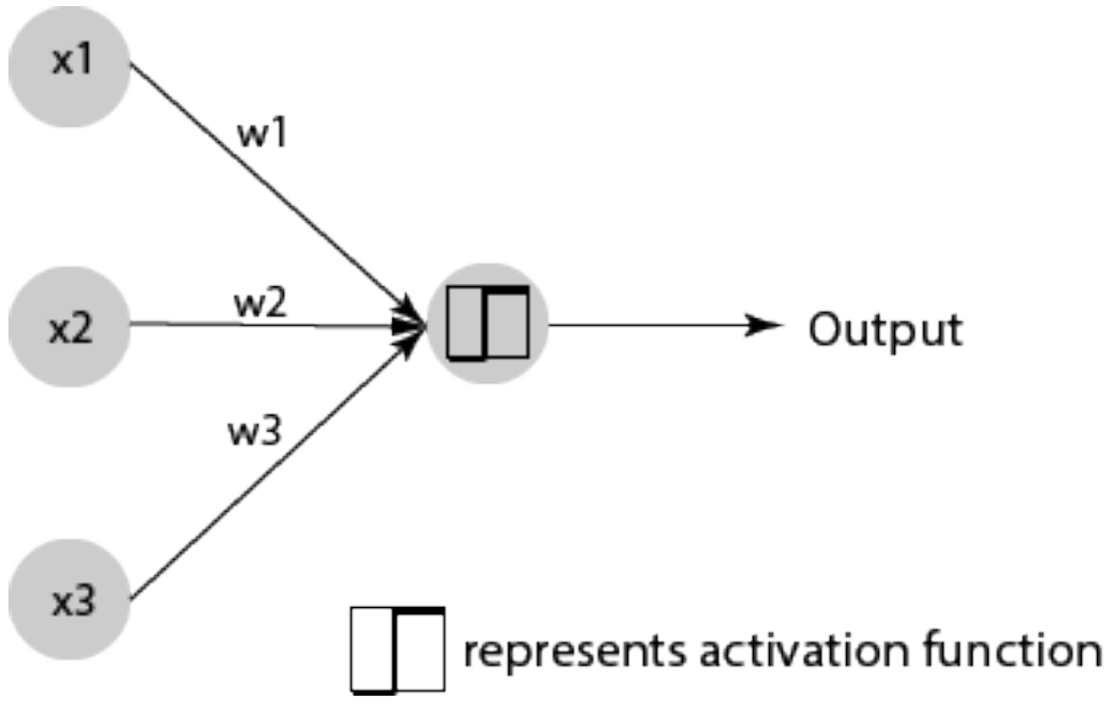
https://ai2-s2-public.s3.amazonaws.com/figures/2016-11-08/3714d0793f5e152ad3c781789f3a1c30f9ef6ee3/0-Figure1-1.png

Classification with Neural Network

Neural Network

- Perceptron

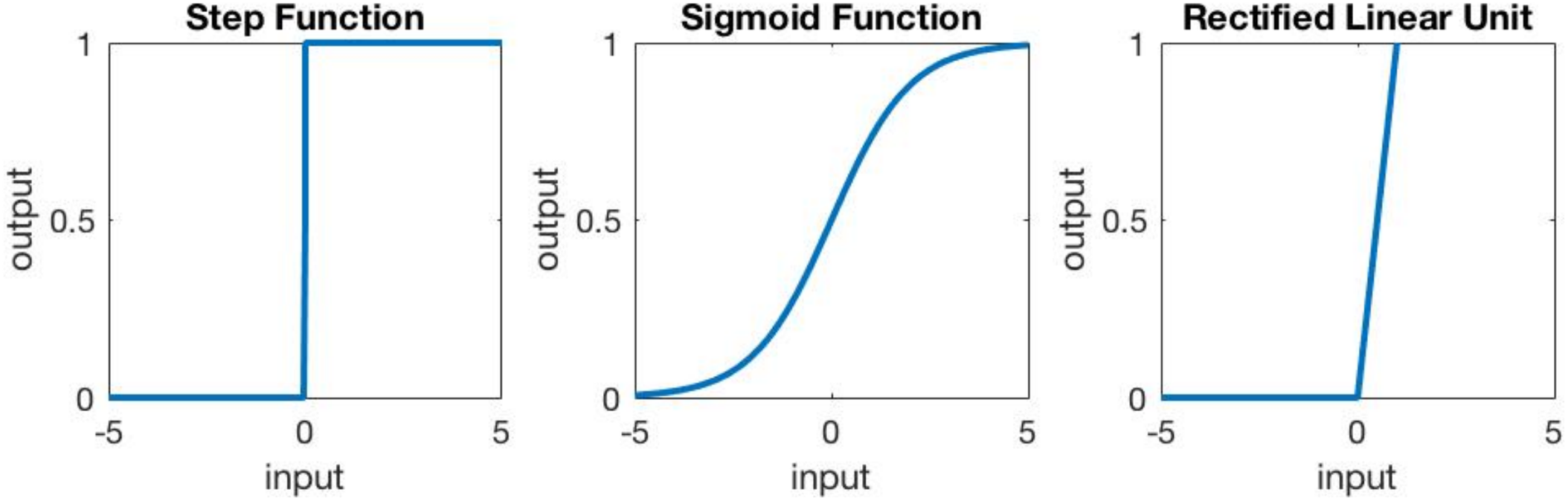
$$output = \begin{cases} 0 & \sum_j w_j x_j \leq threshold \\ 1 & \sum_j w_j x_j > threshold \end{cases}$$



Neural Network

- Activation Function

$$f(z) = \begin{cases} 0 & \sum_j w_j x_j + b_j \leq 0 \\ 1 & \sum_j w_j x_j + b_j > 0 \end{cases} \qquad f(z) = \frac{1}{1+e^{-z}} \qquad f(z) = max(0, x)$$

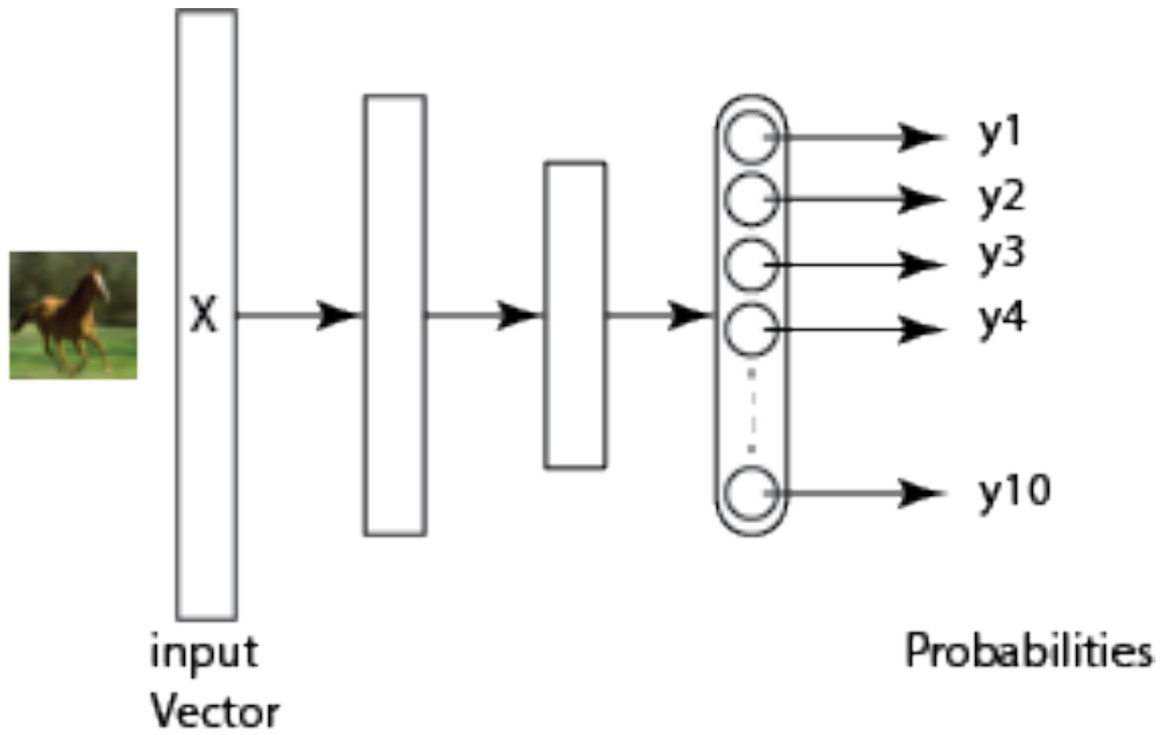


Neural Network

$$u_k = W \cdot z + b$$

- Using Softmax function as multi-class classifier

$$y_k = \frac{exp(u_k)}{\sum_{j=1}^{K} exp(u_j)}$$



- Maximum likelihood estimation and logarithmic transformation were taken to get the error function

$$E(w) = -\sum_{n=1}^{N}\sum_{k=1}^{K} d_{nk} \log y_k(x_n; w)$$

▪▪▪ Cross Entropy

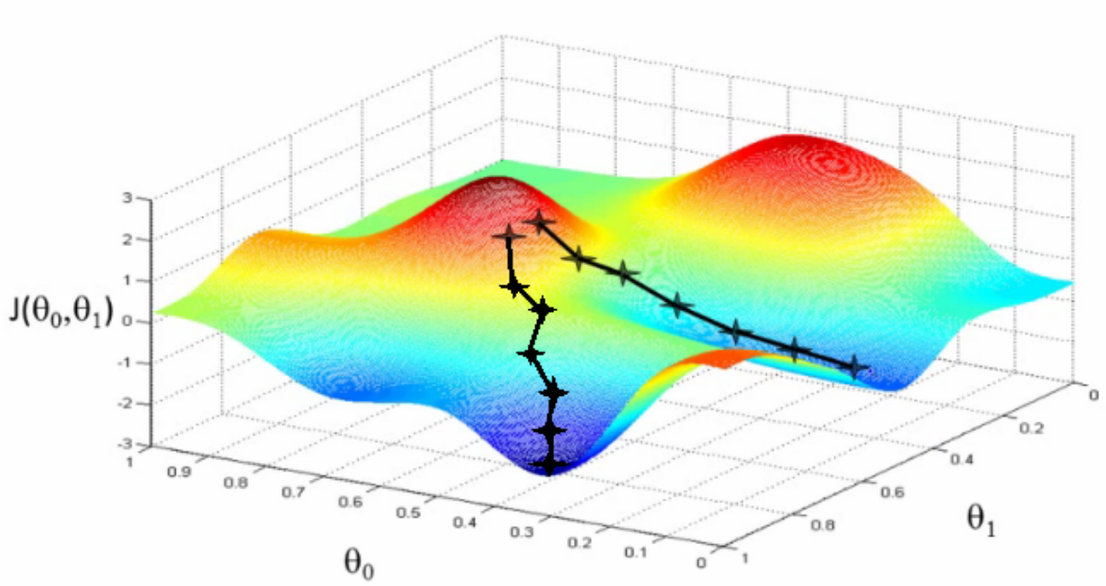
Neural Network

$$E(w) = -\sum_{n=1}^{N}\sum_{k=1}^{K} d_{nk} \log y_k(x_n; w)$$

$$\nabla E = \frac{\partial E}{\partial W} = \left[\frac{\partial E}{\partial w_1} \ \frac{\partial E}{\partial w_2} \cdots \frac{\partial E}{\partial w_M}\right]^T$$

$$W^{t+1} = W^t - \epsilon \nabla E$$

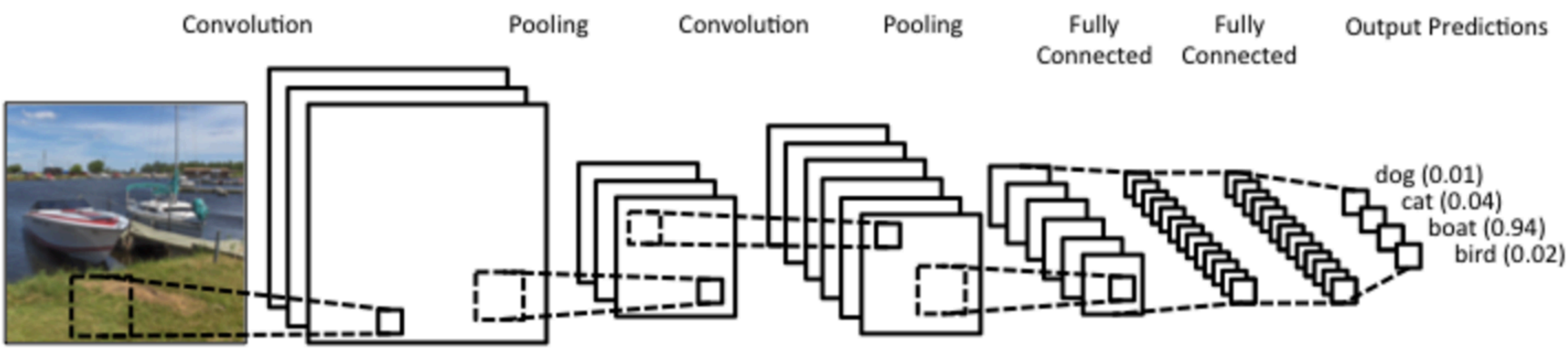
$$W^{t+1} = W^t - \epsilon \nabla E$$



http://blog.datumbox.com/wp-content/uploads/2013/10/gradient-descent.png

Convolutional Neural Network(CNN)

Constructed with Convolution and Pooling Operations

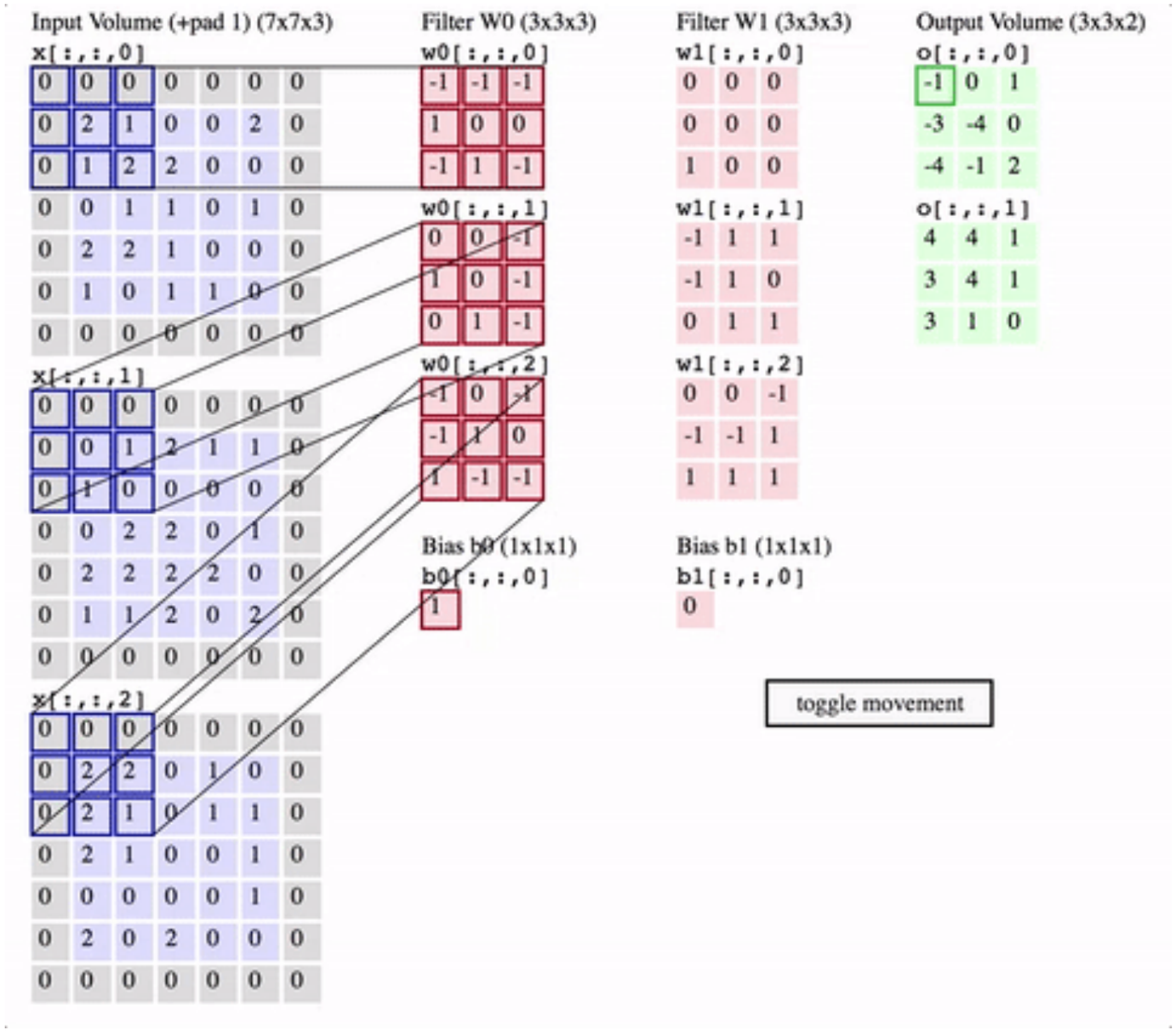


http://d3kbpzbmcynnmx.cloudfront.net/wp-content/uploads/2015/11/Screen-Shot-2015-11-07-at-7.26.20-AM.png

Convolutional Neural Network(CNN)

1. Convolution

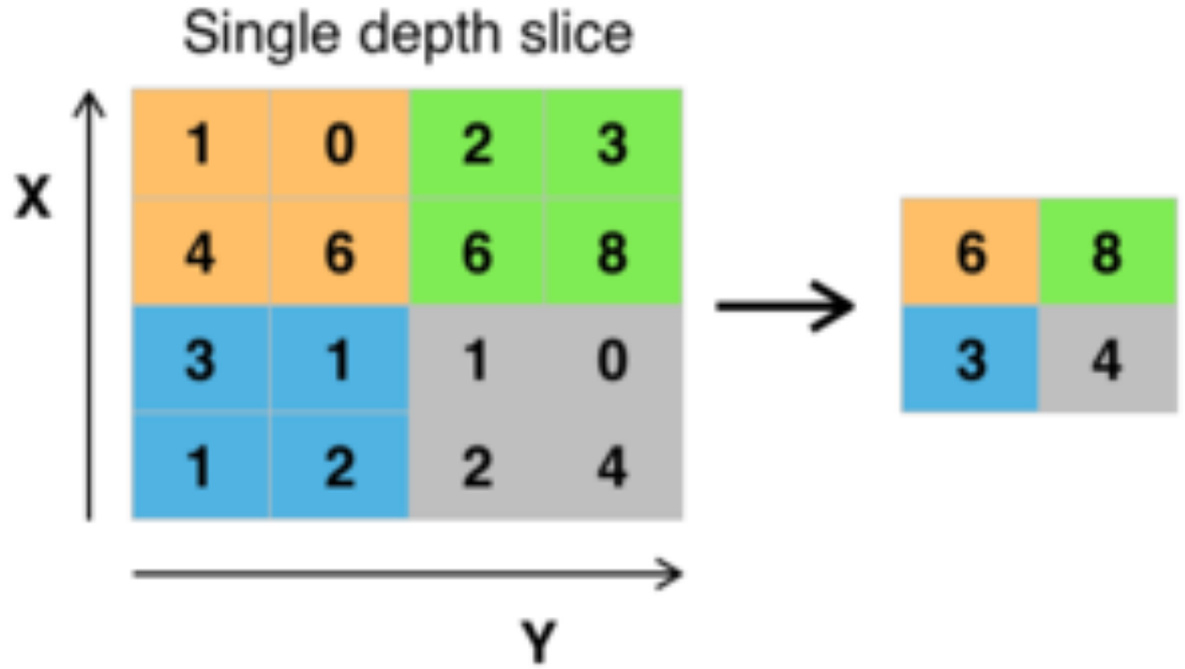
- Locally Connected
- Spacial Correlation are preserved
- Weight Sharing



Convolutional Neural Network(CNN)

2. Pooling

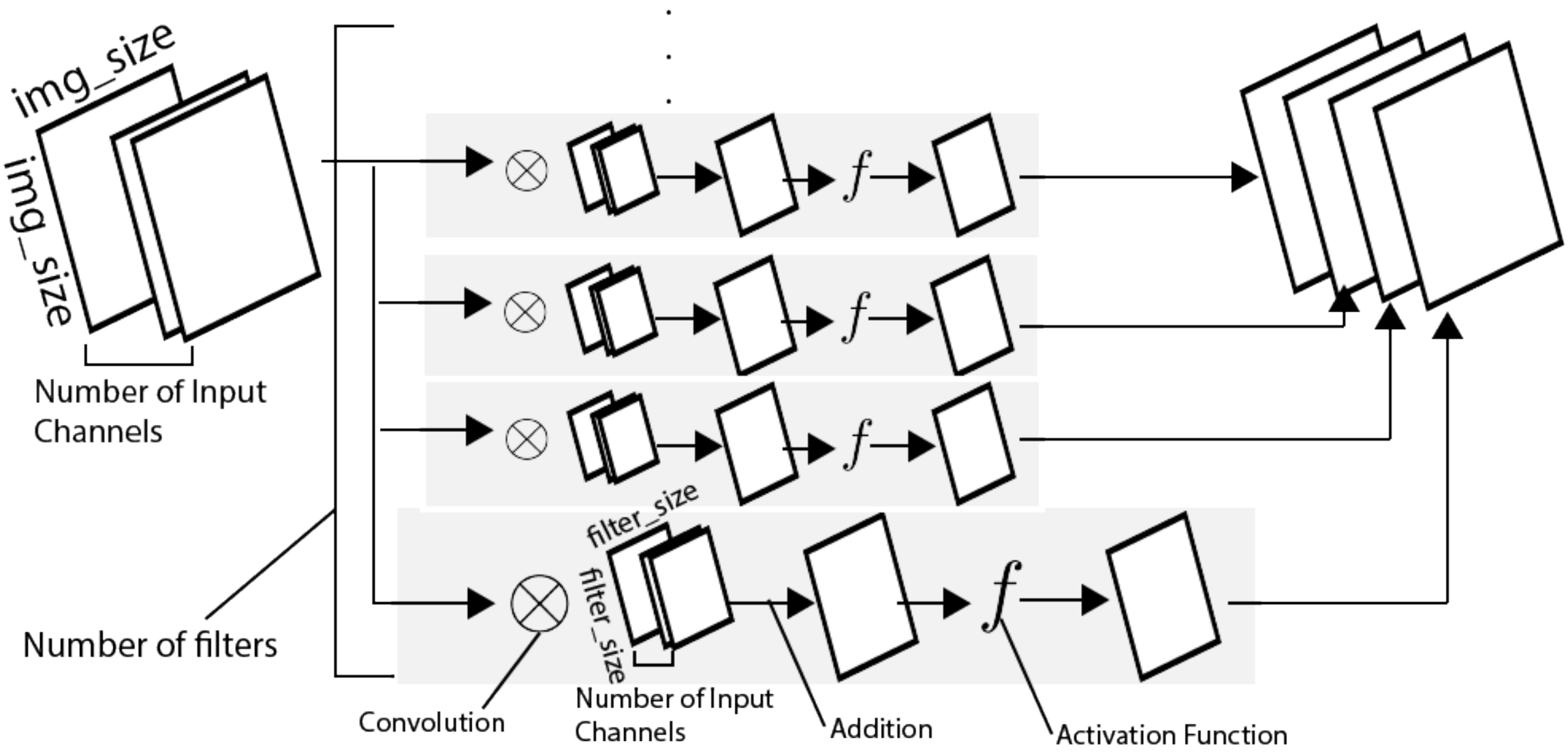
- Subsampling Image
- Dimension reduction



https://upload.wikimedia.org/wikipedia/commons/thumb/e/e9/Max_pooling.png/314px-Max_pooling.png

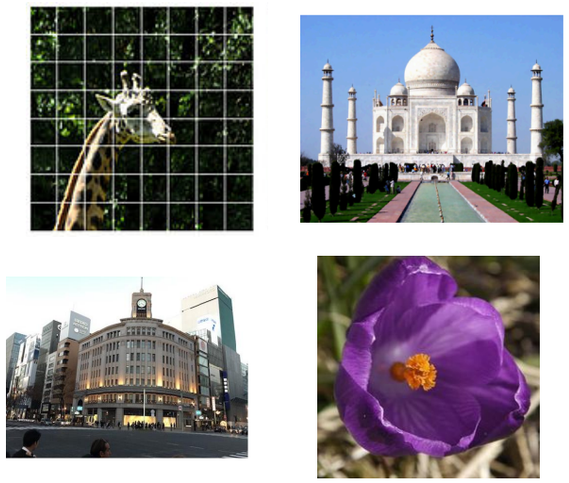
Convolutional Neural Network

3. Multiple Channels, Filters



Our Model

Raw Images(Existing Method)



Classification with Neural Network

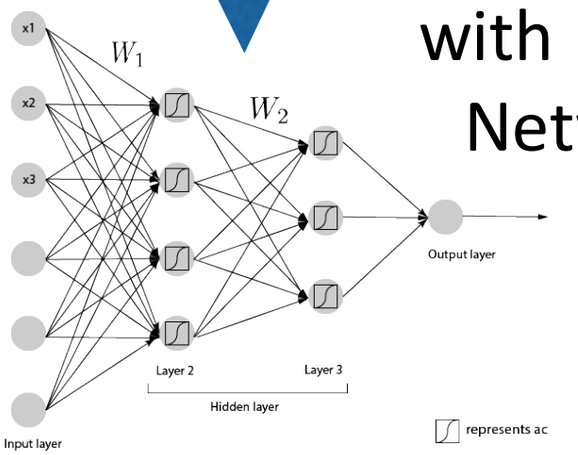
Feature Enhanced images (Our Method)

Color Layout Descriptor

Anisotropic Diffusion

Canny Edge

Ultrametric Contour Map



Classification with Neural Network

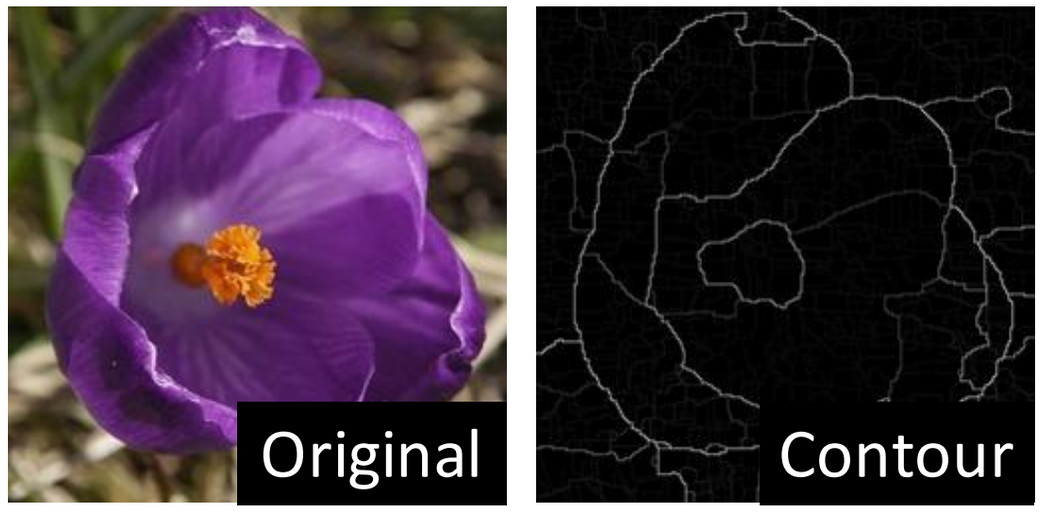
Implementation

CPU	Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz
Memory	12GB
GPU	GeForce GTX 1060 ARGMOR 6G OC by msi
Storage	120GB SSD + 500 GB HDD
Software	Matlab 2016a, CUDA tool-box 8.0, Tensorflow r0.12
OS	Ubuntu 16.10

Extracting Features

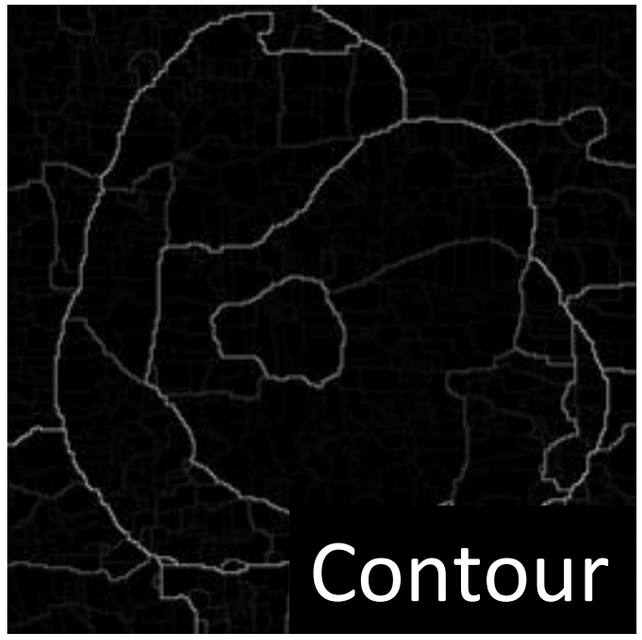
- Anisotropic Diffusion

- Ultrametric Contour Map



Combining Features

- We want to keep the inner color information of object, so we combine the contour by giving Saturation or Brightness on the original images.



Constructing Neural Network

- A open source software library for conducting machine learning tasks.

- Released on late 2015 by Google Brain

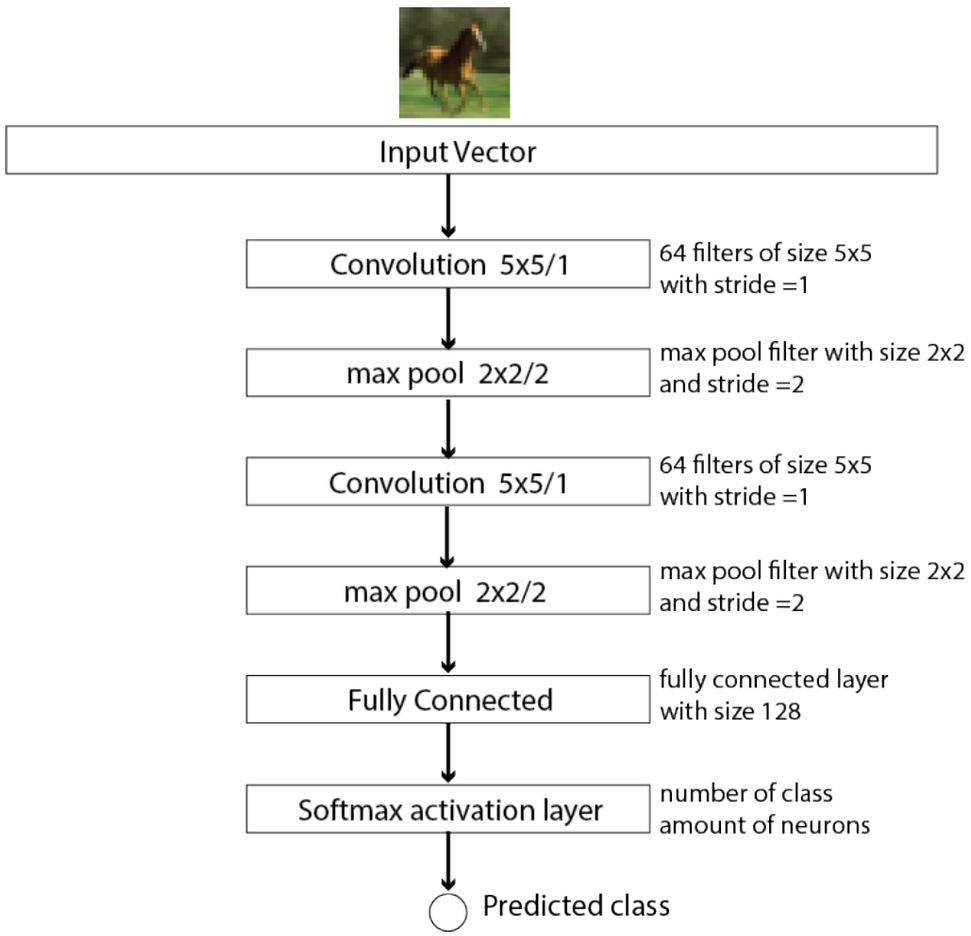
- Python,C++ as API

- GPU supports without explicitly program parallel code

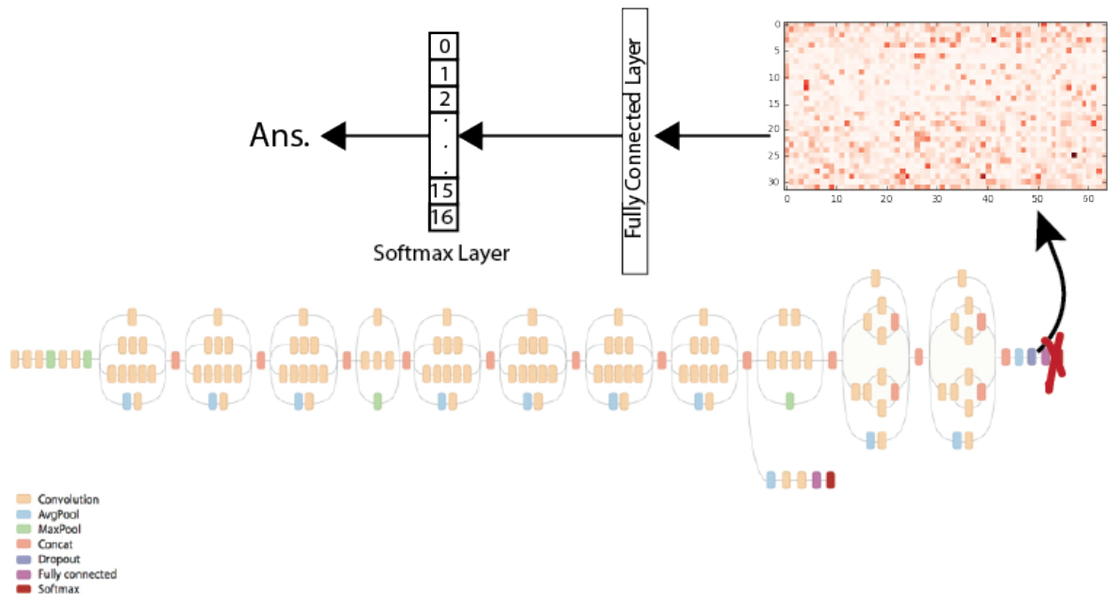
https://i.ytimg.com/vi/oZikw5k_2FM/maxresdefault.jpg

Constructing Neural Network

My Implementation

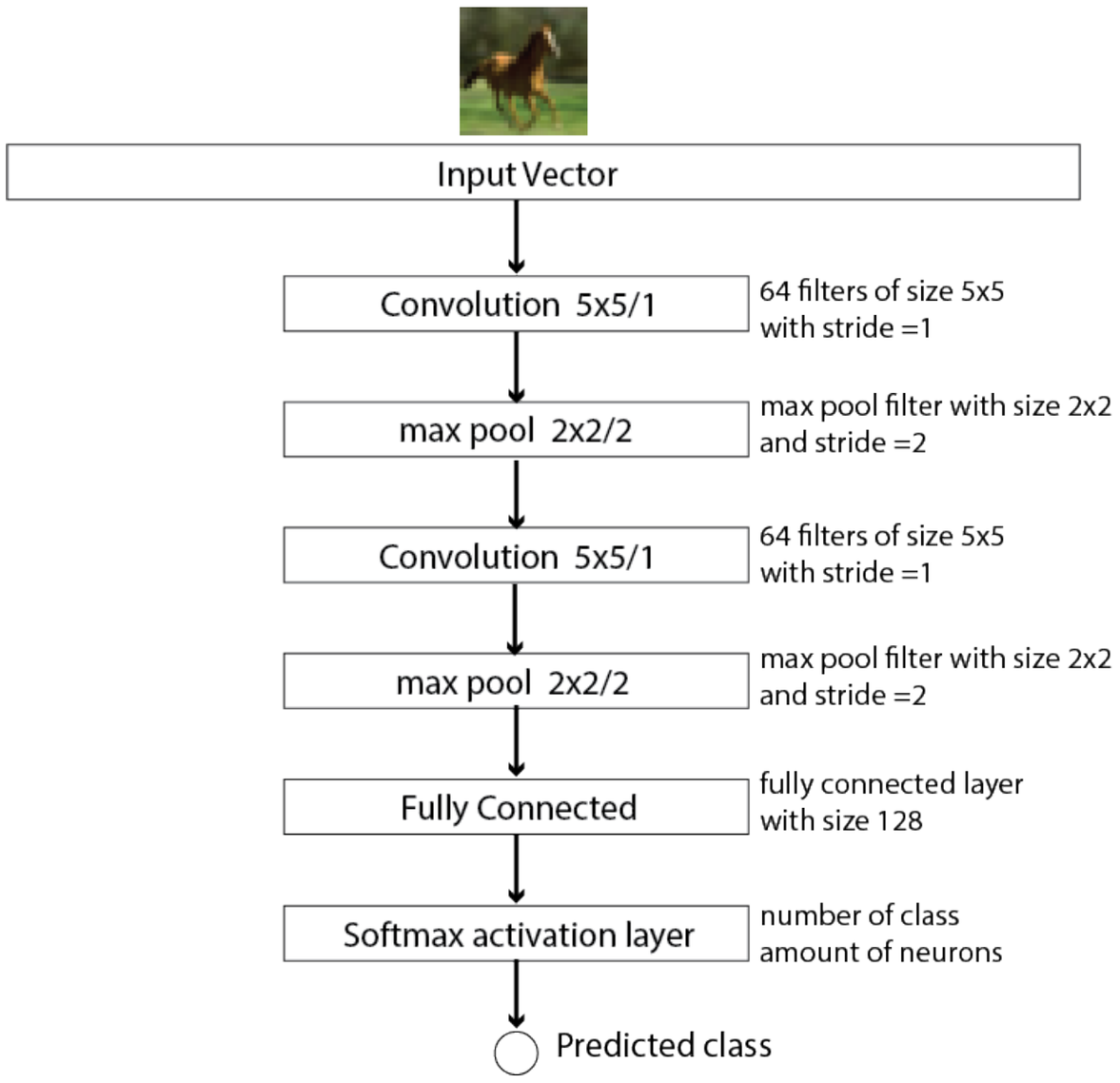


Transfer Learning with Inception Model



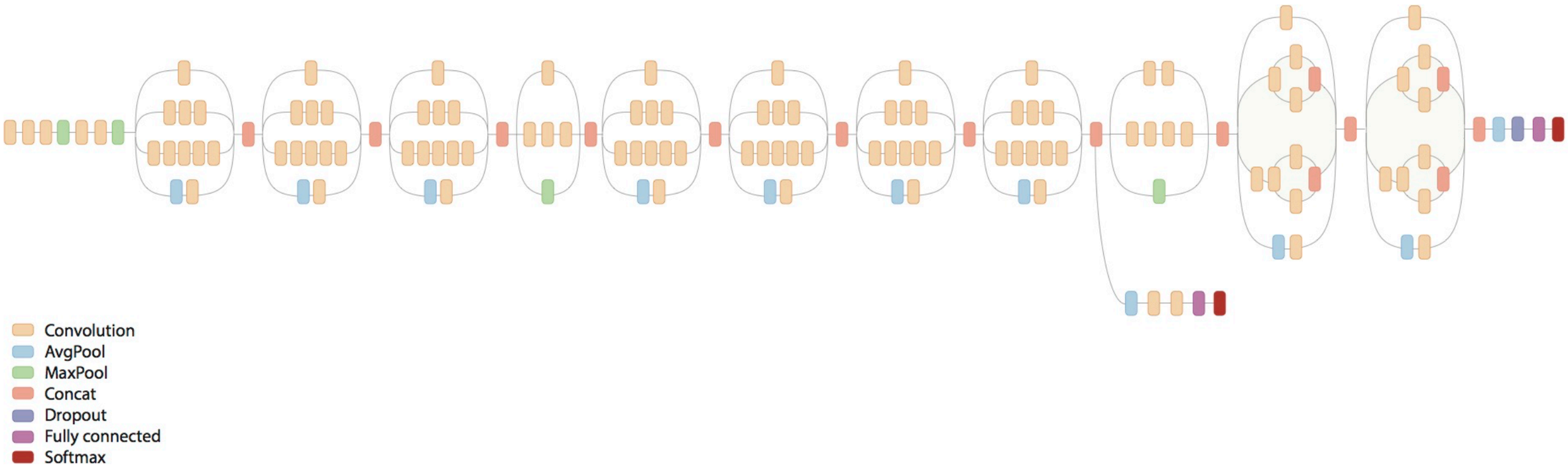
My Implementation

- 2 Convolution, 1 Fully connected Layer.
- Advantage of setting unique initial variables
- Can see the weight and output value at each layer
- Not accurate for practical use.



Inception Model

- It is Google's trained network "Inception-v3" trained with 1000 categories of objects over millions of images

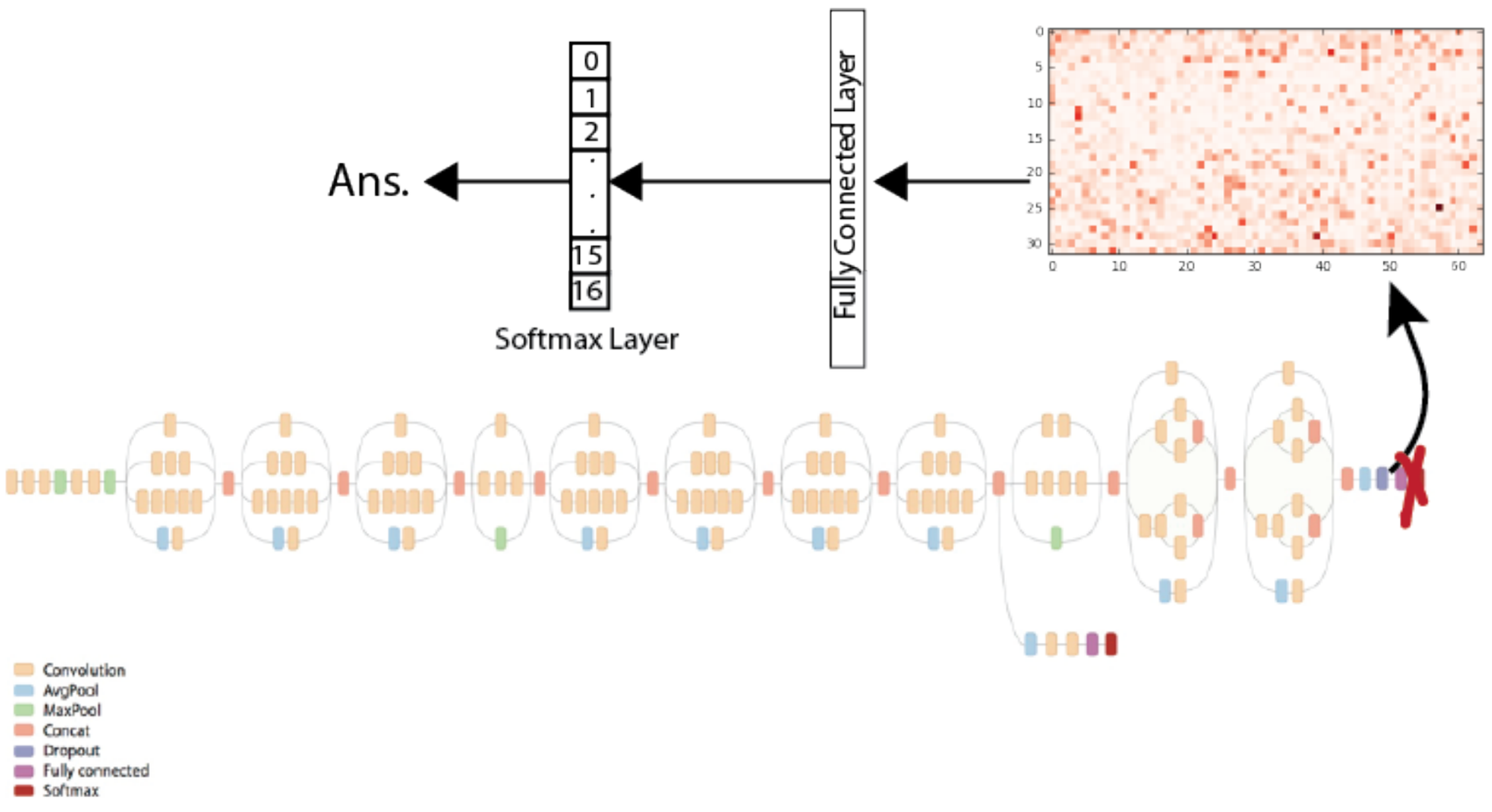


https://4.bp.blogspot.com/-TMOLlkJBxms/Vt3HQXpE2cI/AAAAAAAAA8E/7X7XRFOY6Xo/s1600/image03.png

Transfer Learning with Inception Model

- Adding few new classification layer in the end, the network outputs practical level of high accuracy.

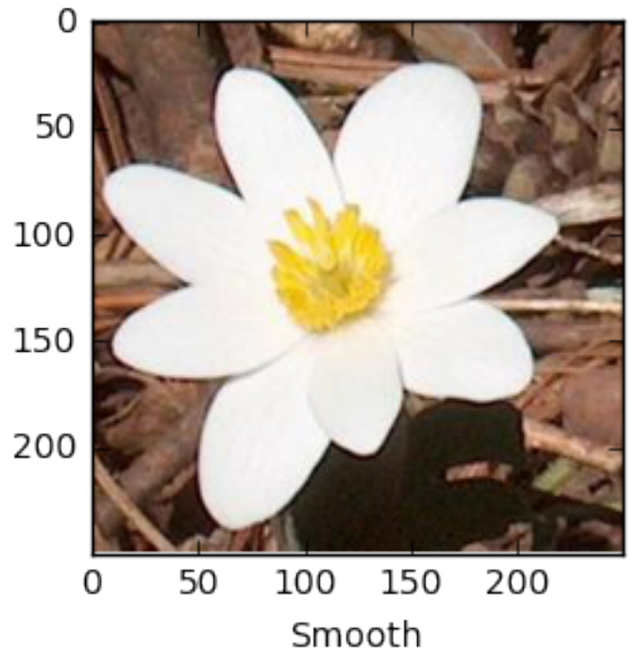
- Can NOT optimize the parameter inside the network because it is computationally heavy.



Experiments

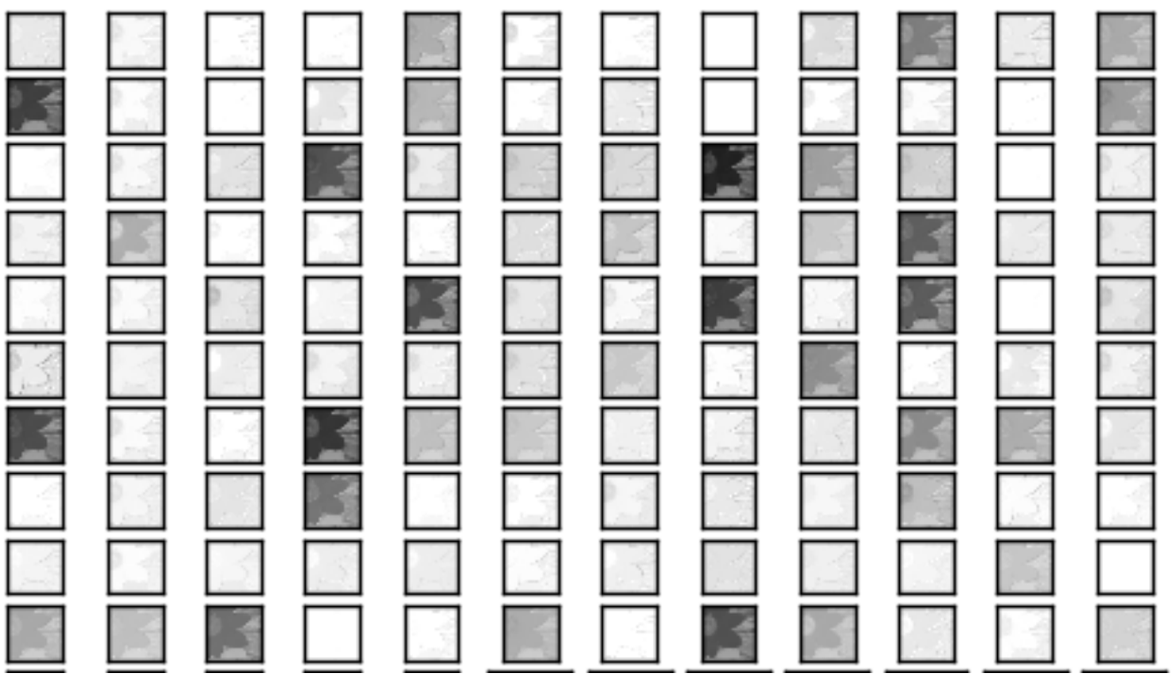
Effect on Neural Network

- **Input Image example**

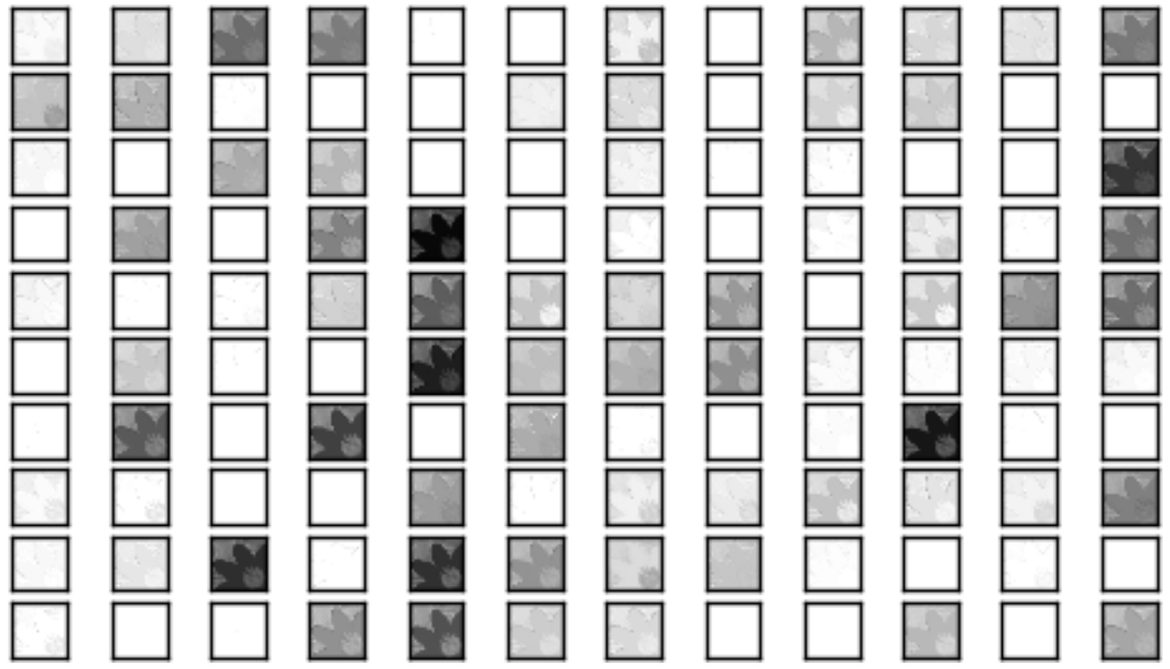


Effect on Neural Network

- **1st convolution layer output**



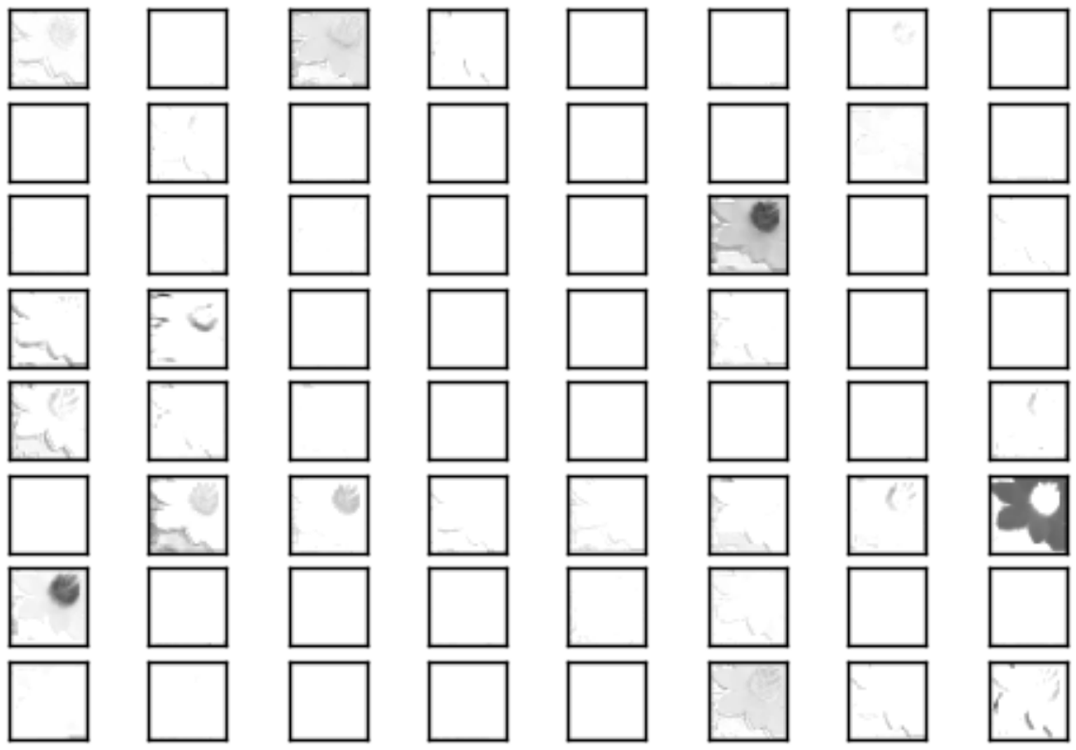
Using Raw Image as input



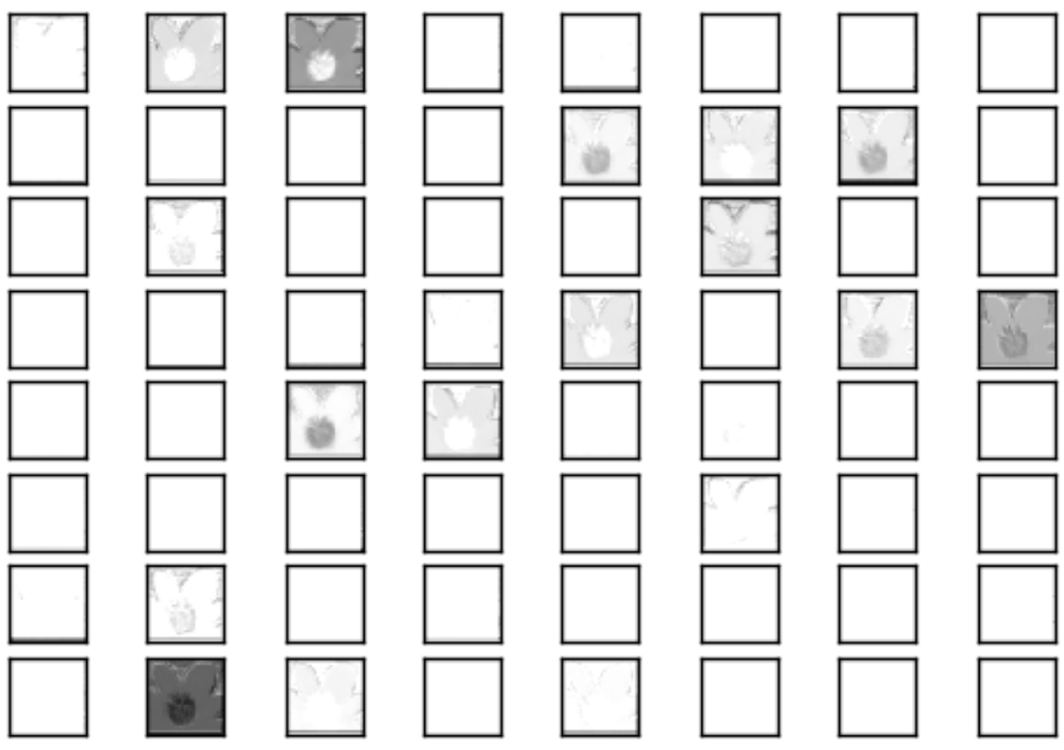
Using Contour Enhanced Image as input

Effect on Neural Network

- **2nd convolution layer output**



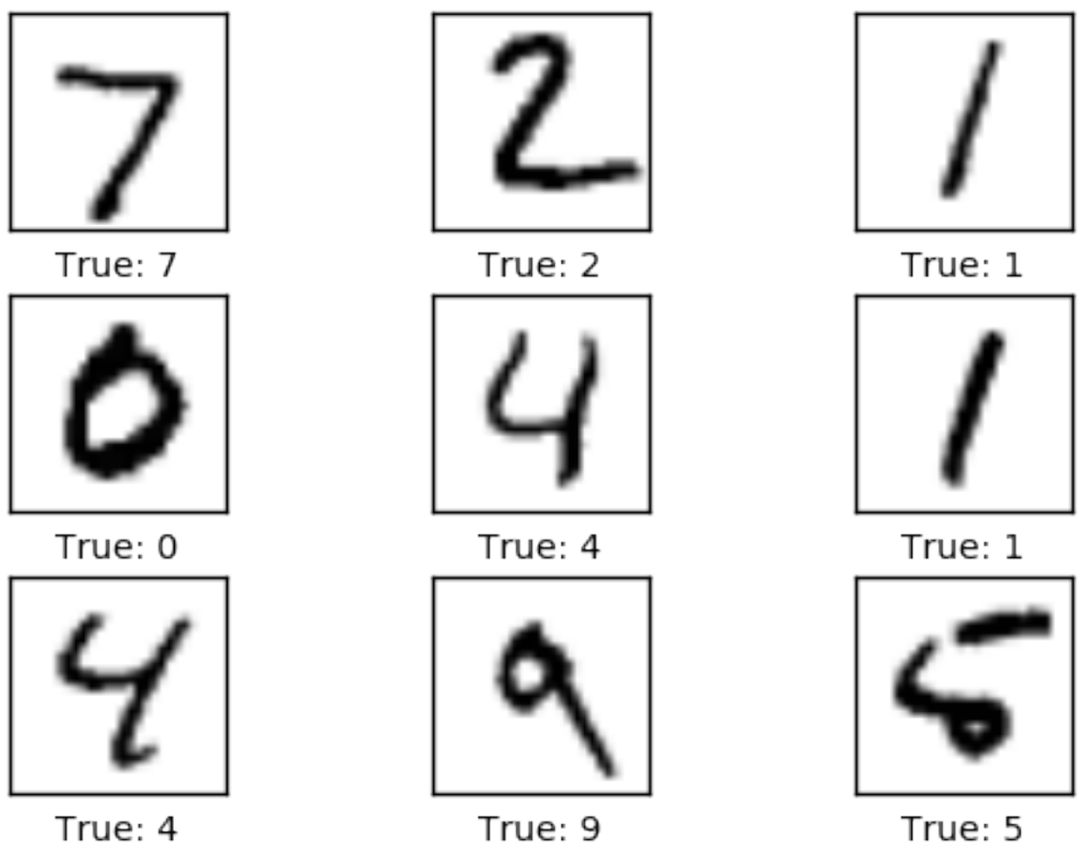
Using Raw Image as input



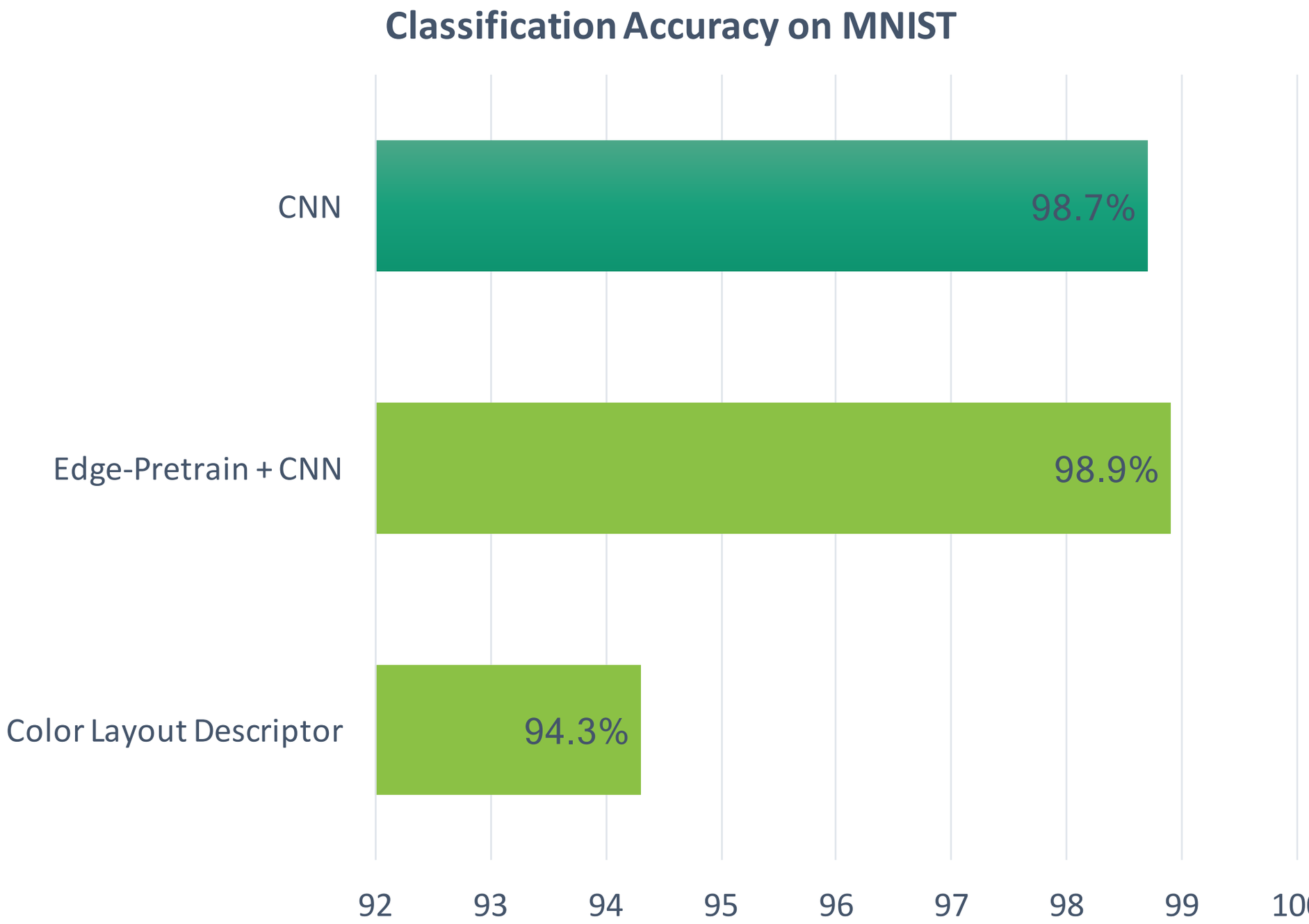
Using Contour Enhanced Image as input

MNIST Experiments

- handwritten digits of number between 0 to 9
- Each image is 28x28 pixels
- 60,000 training images and 10,000 testing images

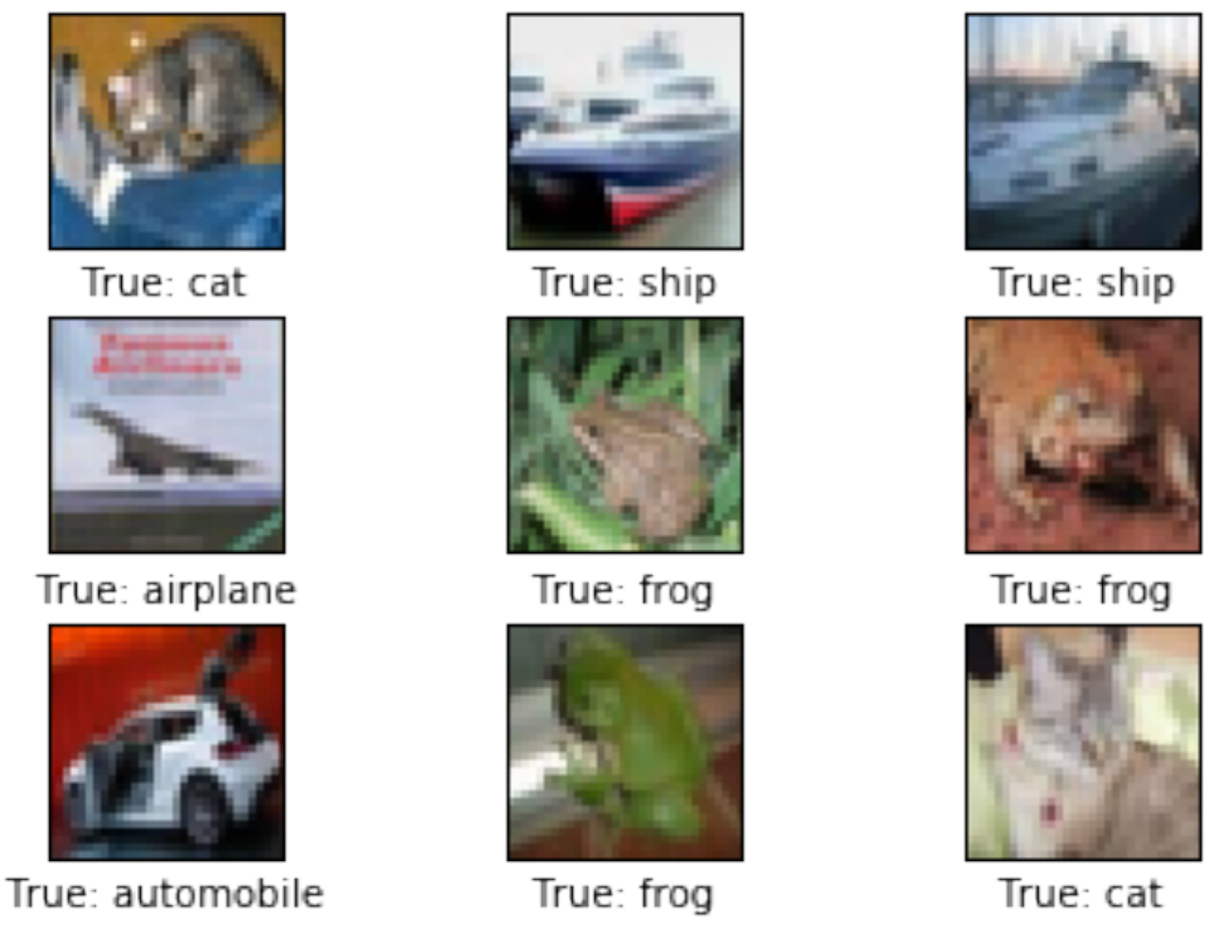


MNIST Experiments

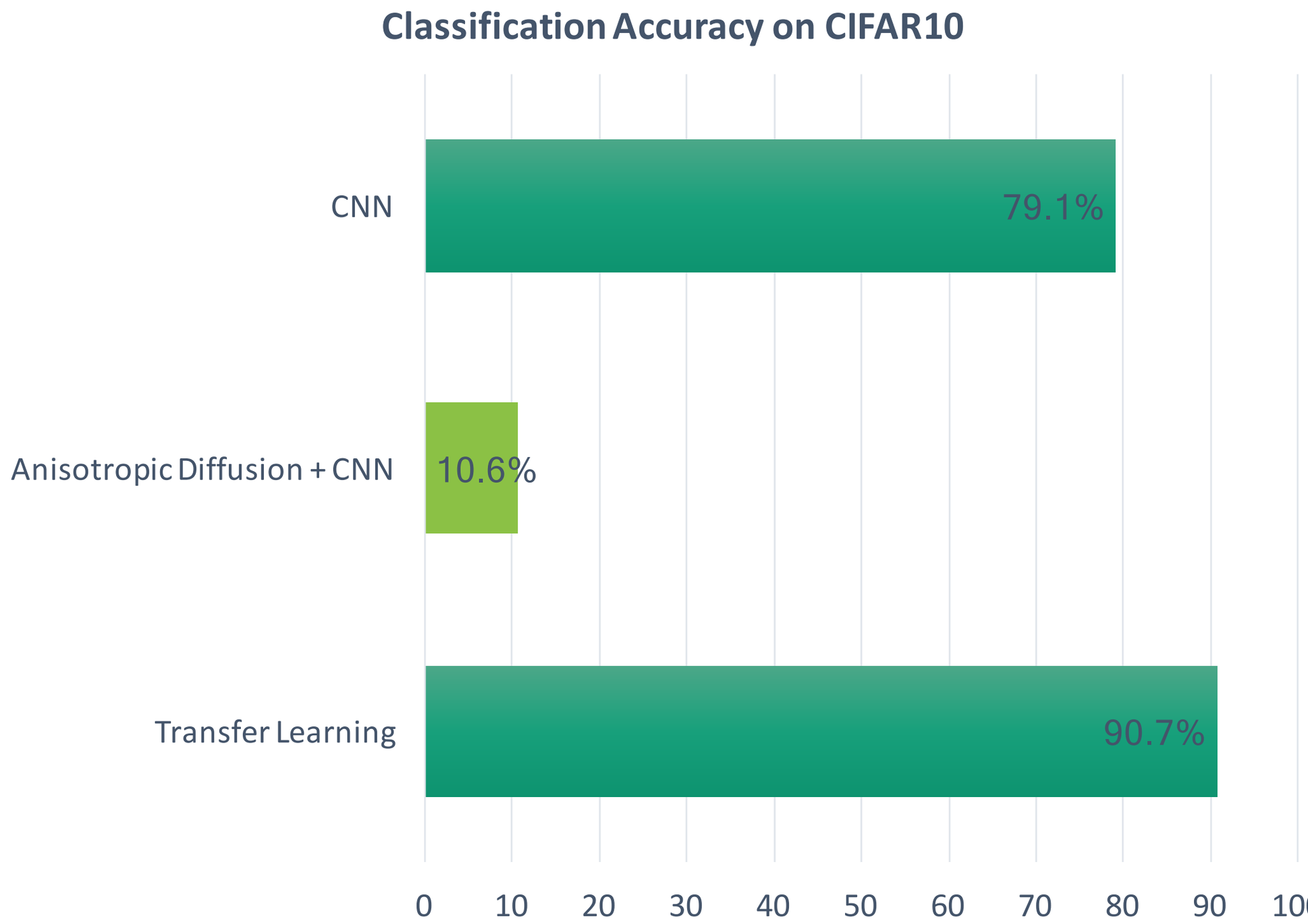


CIFAR10 Experiments

- 10 object classes.
- Each image is a natural 32x32 colored image.
- 50,000 training images and 10,000 test images.



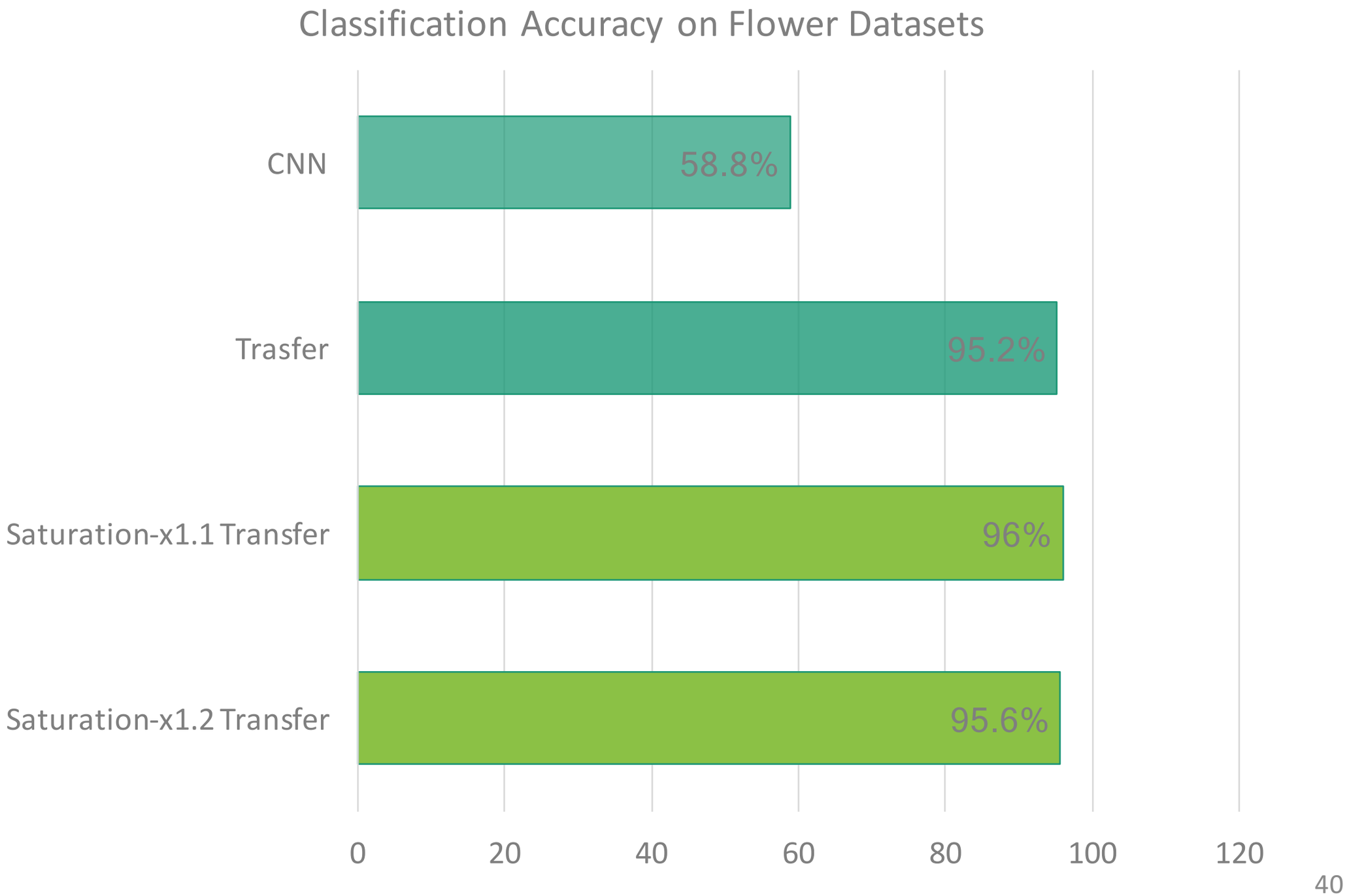
CIFAR10 Experiments



Oxford Flower Datasets Experiments

- 17 Flower Classes
- 1360 images
- 1088 images for training and the rest 272 images are for testing

Oxford Flower Datasets Experiments



Conclusion

- The contour or shape of the object are indeed strong feature. Even stronger than Color.
- Enhancing contour through saturation value are effective at making neural network to focus those feature and perform better at classification.

Thank you!