# Vehicle Identification and Classification System

Institute of Engineering and Technology, Lucknow

Information Technology Program
(A self-financed course)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vishal Polley (1505213053)

Supervised By -
Ms. Shipra Gautam

# Problem Statement

- Implementation of an efficient method for recognizing vehicles.
- Classification of objects e.g. as cars, trucks, pedestrians, etc. in the traffic dataset.

# Solution

1. Bag of Features Classifier

    - Implementation of feature extraction module using bag of features (combination of Harris-corner detector and SIFT features)
    - Classification is performed using Support vector machines (SVM)

2. Deep Learning Classifier
    - Convolutional Neural Network

# Image Dataset

1. Indian Vehicle Database


| Properties | Description |
|---|---|
| Name | Indian Vehicle database |
| Sources | Static vehicle pictures captured using camera on Indian roads, Pictures collected from Internet resources like Goggle images etc. and Pictures cropped from a traffic videos |
| Constraints | Pose, lightning and view |
| Number of classes | 4 |
| vehicle types | Truck, Auto, Bus and Car |
| Number of images per class | 450 |
| Total Images | 1800 |

# Image Dataset Cont.

2. MIO-TCD Classification Challenge Dataset

| Category | Training | Testing |
|---|---|---|
| Articulated Truck | 10,346 | 2,587 |
| Bicycle | 2,284 | 571 |
| Bus | 10,316 | 2,579 |
| Car | 260,518 | 65,131 |
| Motorcycle | 1,982 | 495 |
| Non-Motorized Vehicle | 1,751 | 438 |
| Pedestrian | 6,262 | 1,565 |
| Pickup Truck | 50,906 | 12,727 |
| Single-Unit Truck | 5,120 | 1,280 |
| Work Van | 9,679 | 2,422 |
| Background | 160,000 | 40,000 |
| **Total** | **519,164** | **129,795** |

# Feature Extraction

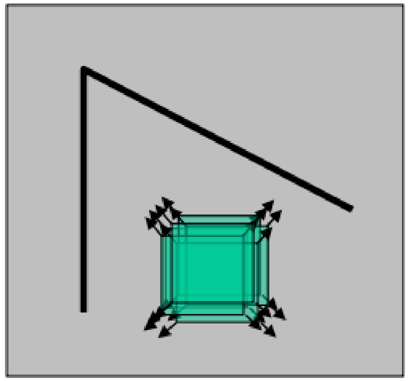
1. Keypoints Detection
2. Computing Descriptors
3. Clustering
4. Bag of Visual Words Model
5. Generating Vocabulary

# Keypoint Detection

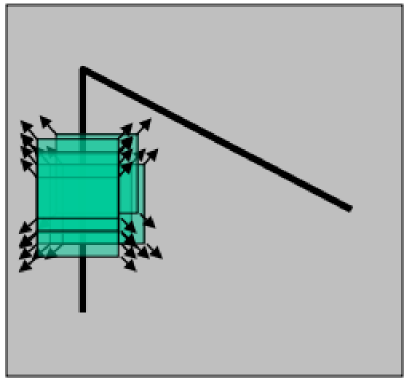
**Harris Corner Detection**

- This algorithm was developed to identify the internal corners of an image.
- The corners of an image are basically identified as the regions in which there are variations in large intensity of the gradient in all possible dimensions and directions.

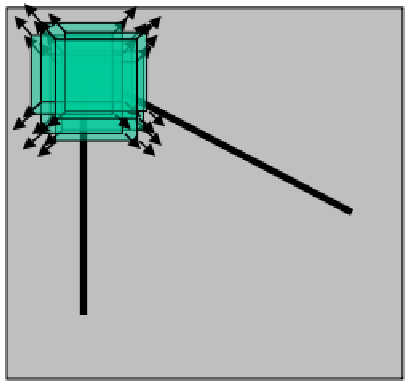
# Harris Corners Detection



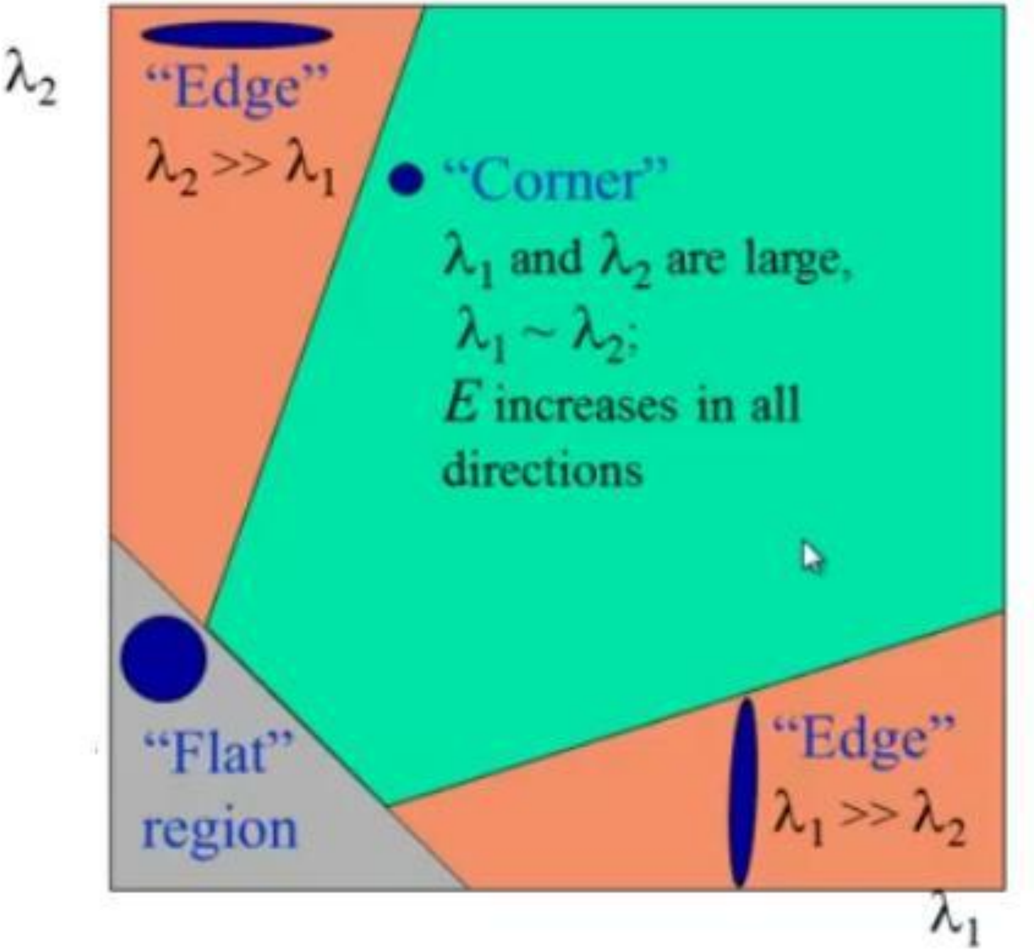
“flat” region:
no change in
all directions



“edge”:
no change along
the edge direction



“corner”:
significant change
in all directions



# Computing Descriptors

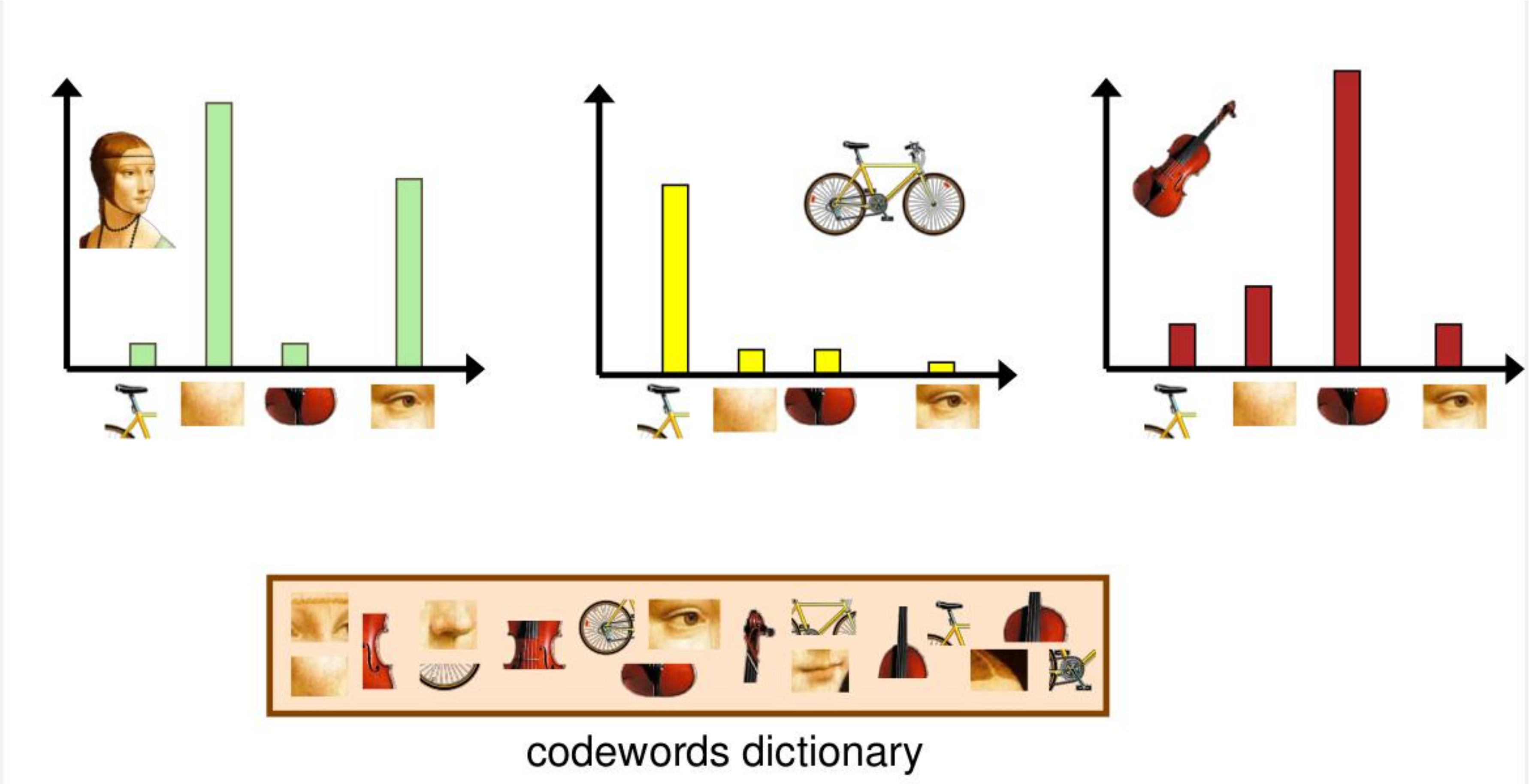
**SIFT (Scale Invariant Feature Transform)**

- It is a technique for detecting salient, stable feature points in an image.
- For every such point, it also provides a set of “features” that “characterize/describe” a small image region around the point.
- These features are invariant to rotation and scale.

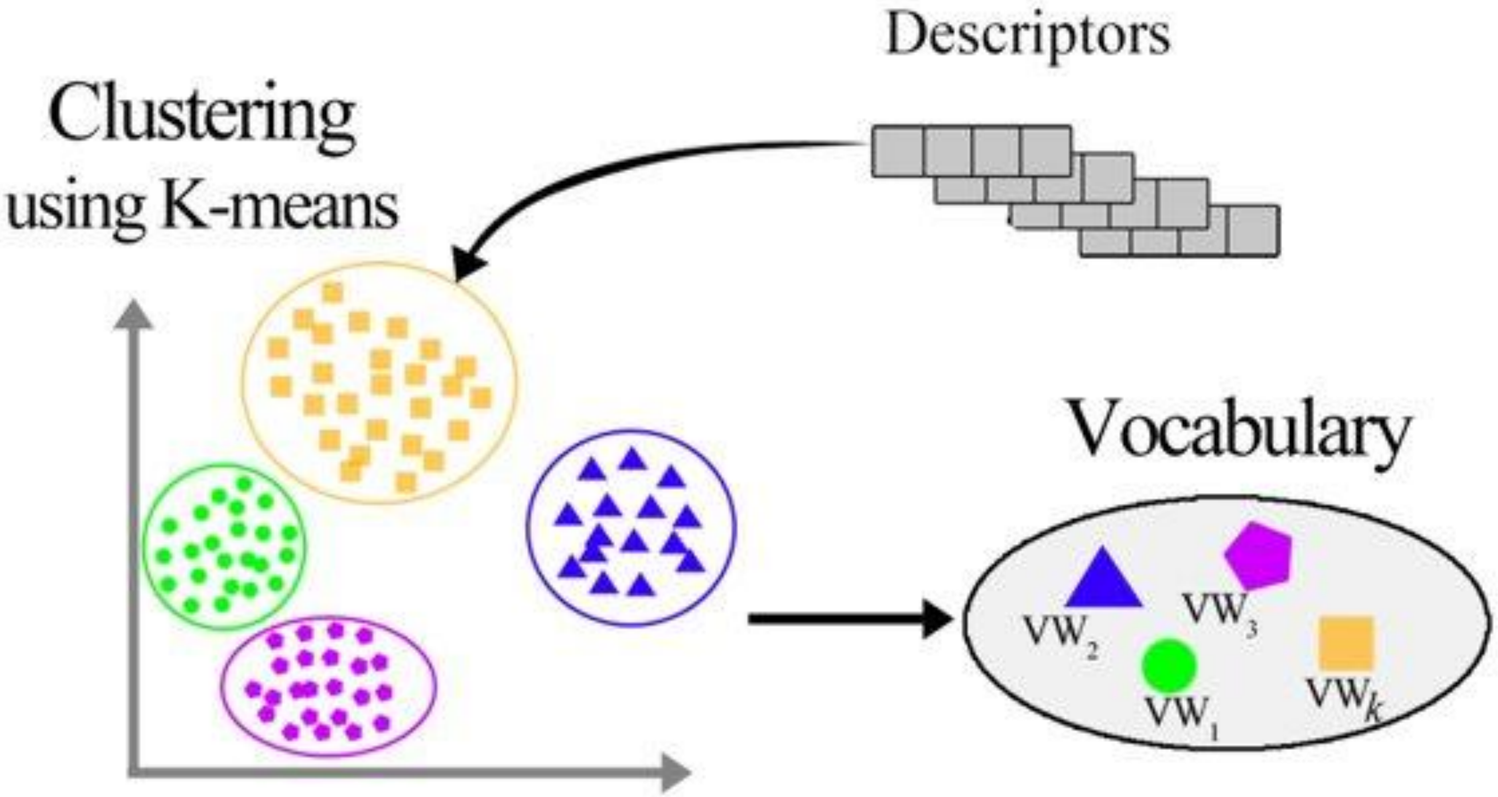
# Bag of Visual Words

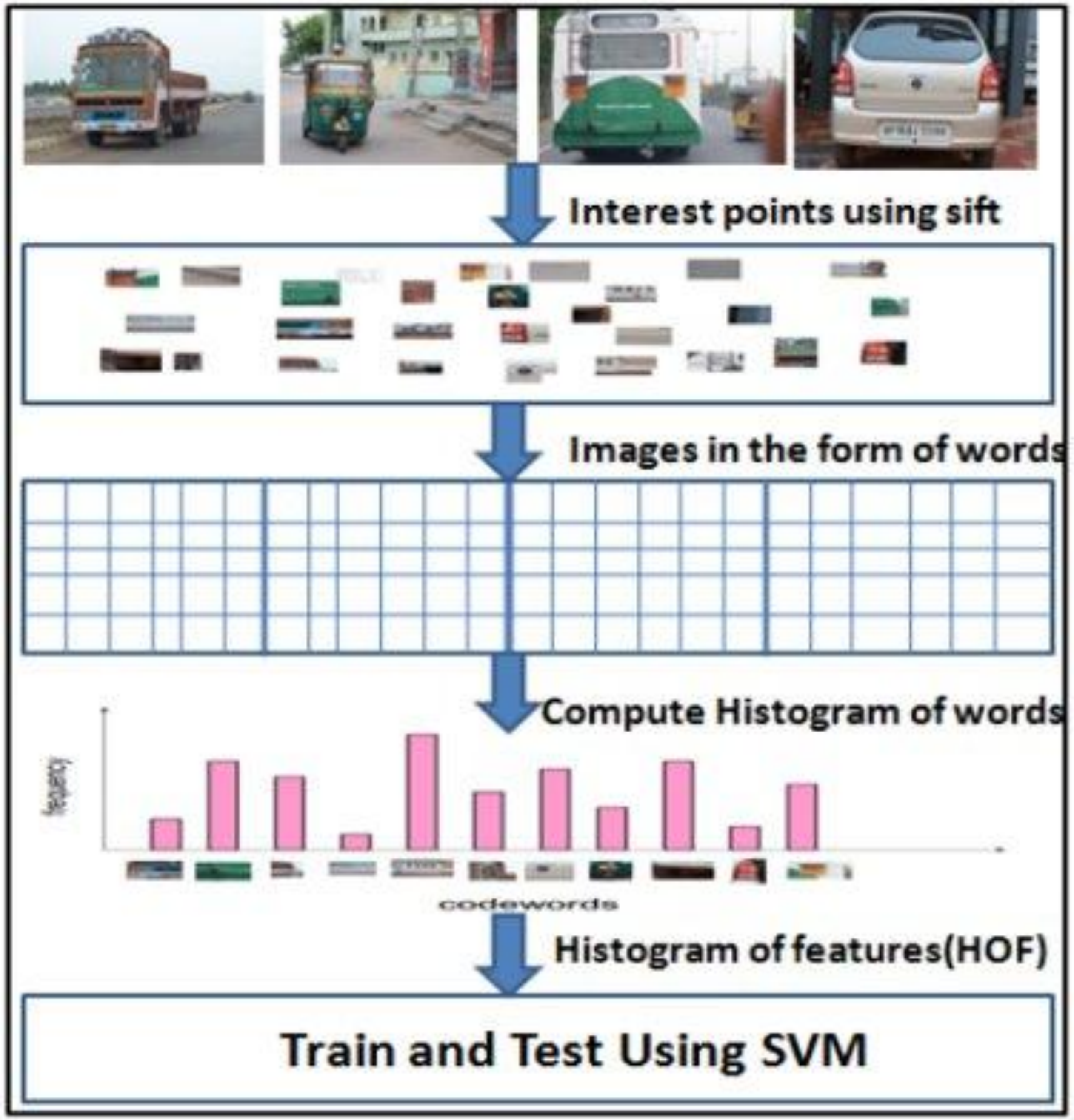
- Supervised Learning model.
- Every object can be represented by its parts.
- A label can be defined as a key/value for identifying to what class/category does the object belongs.
- The final step is codebook generation. A codebook can be thought of as a dictionary that registers corresponding mappings between features and their definition in the object.

# Bag of Visual Words



# Generating Vocabulary

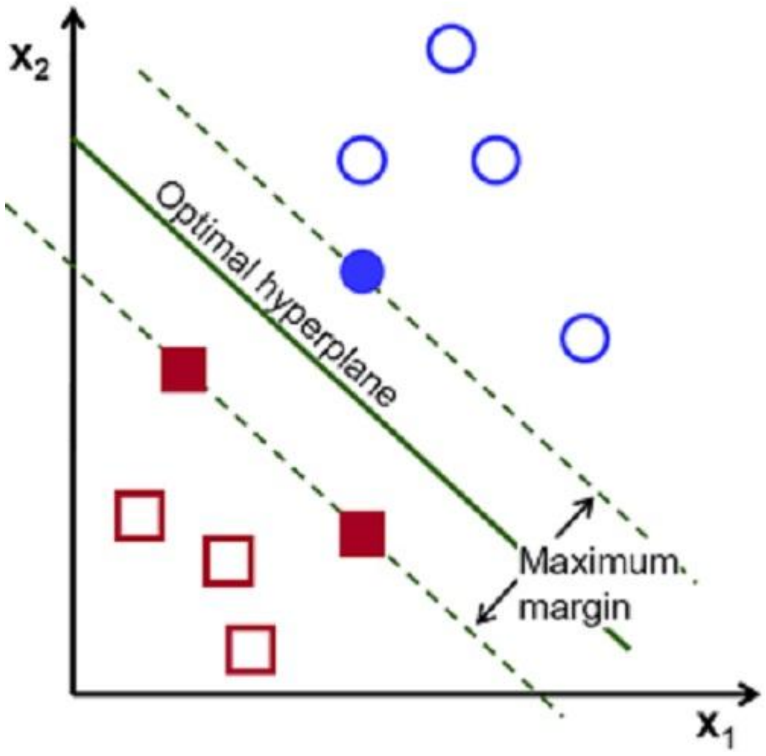



Interest points using sift
Images in the form of words
Compute Histogram of words
frequency
codewords
Histogram of features(HOF)
Train and Test Using SVM

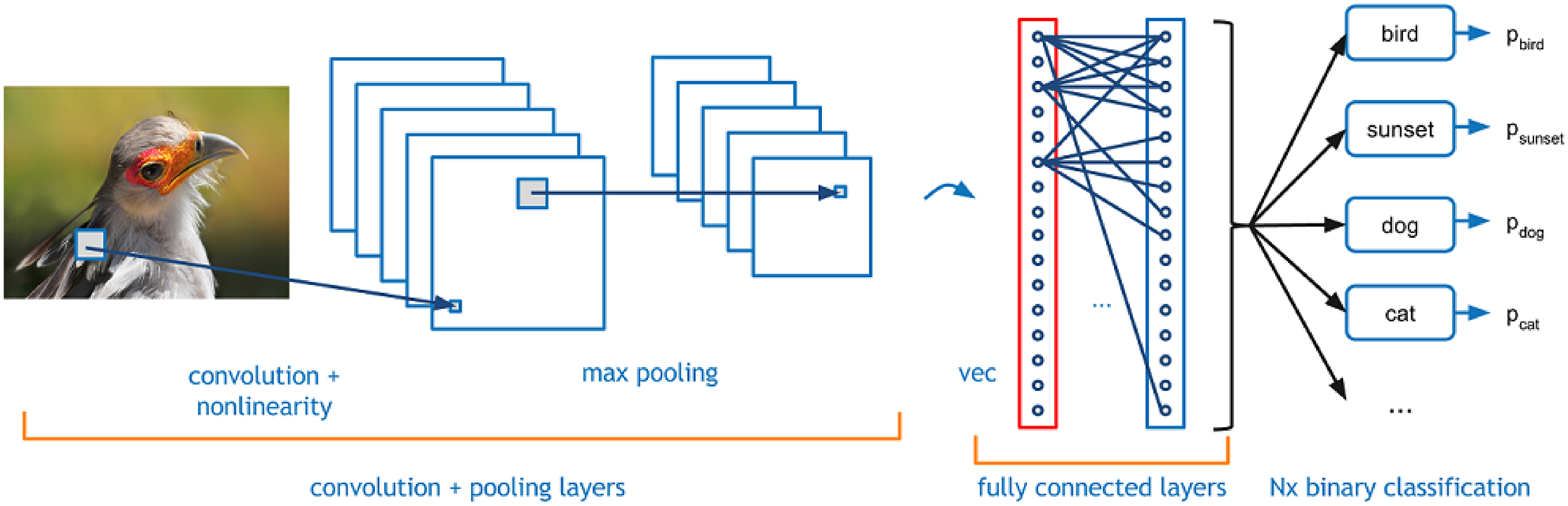
# Classification

**Support Vector Machines (SVM)**

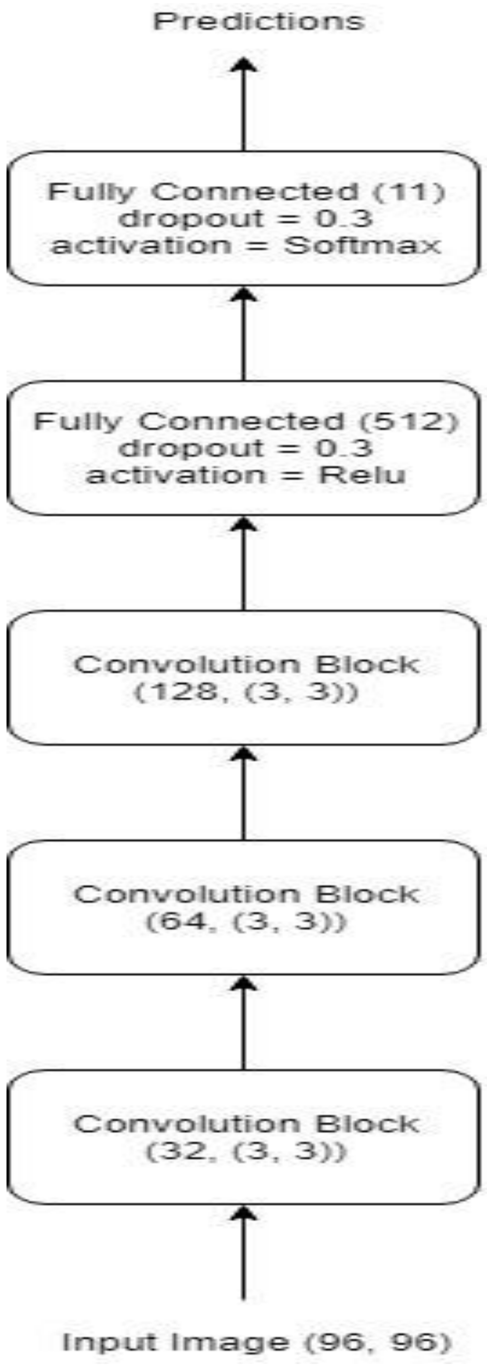
- It is a discriminative classifier formally defined by a separating hyper-plane.
- It is a multiclass classifier to distinguish between similar images and to define classes for the same.



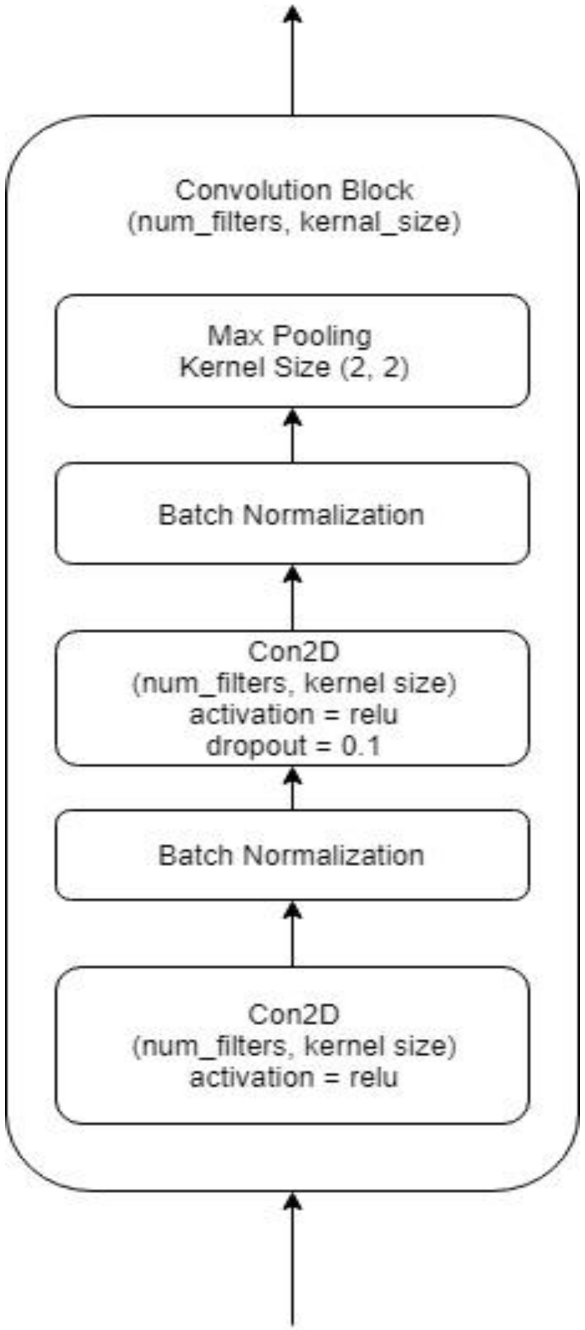
# Convolutional Neural Networks



Architecture of the Deep Learning Model

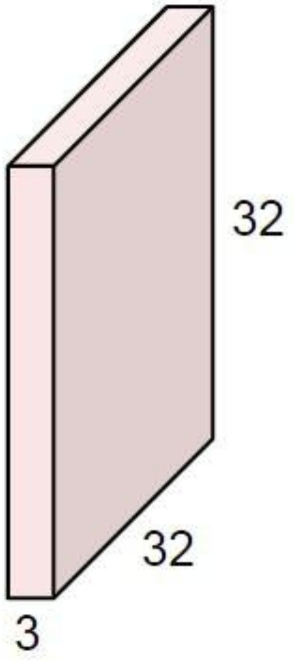


# Modular Architecture of a Convolution Block

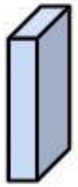


# Convolution Layer

32x32x3 image




5x5x3 filter

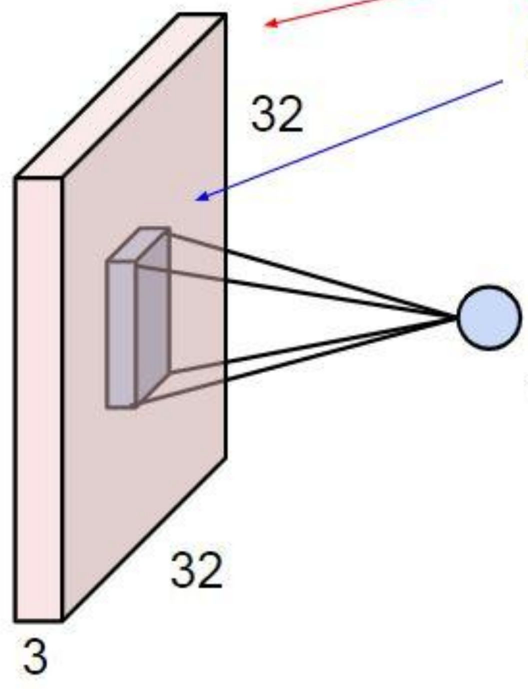


**Convolve** the filter with the image
i.e. “slide over the image spatially,
computing dot products”

# Convolution Layer

32x32x3 image

5x5x3 filter $w$

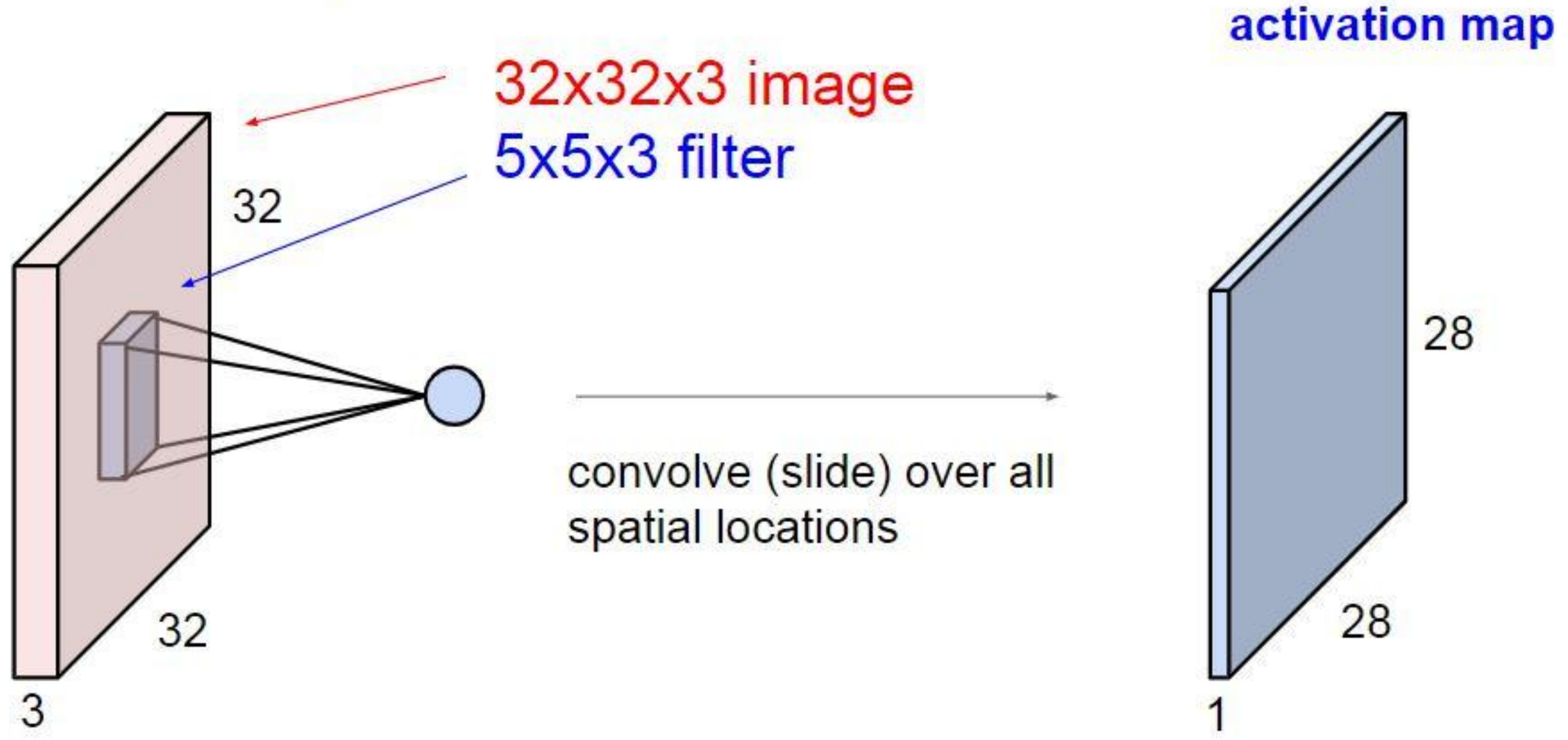



**1 number:**

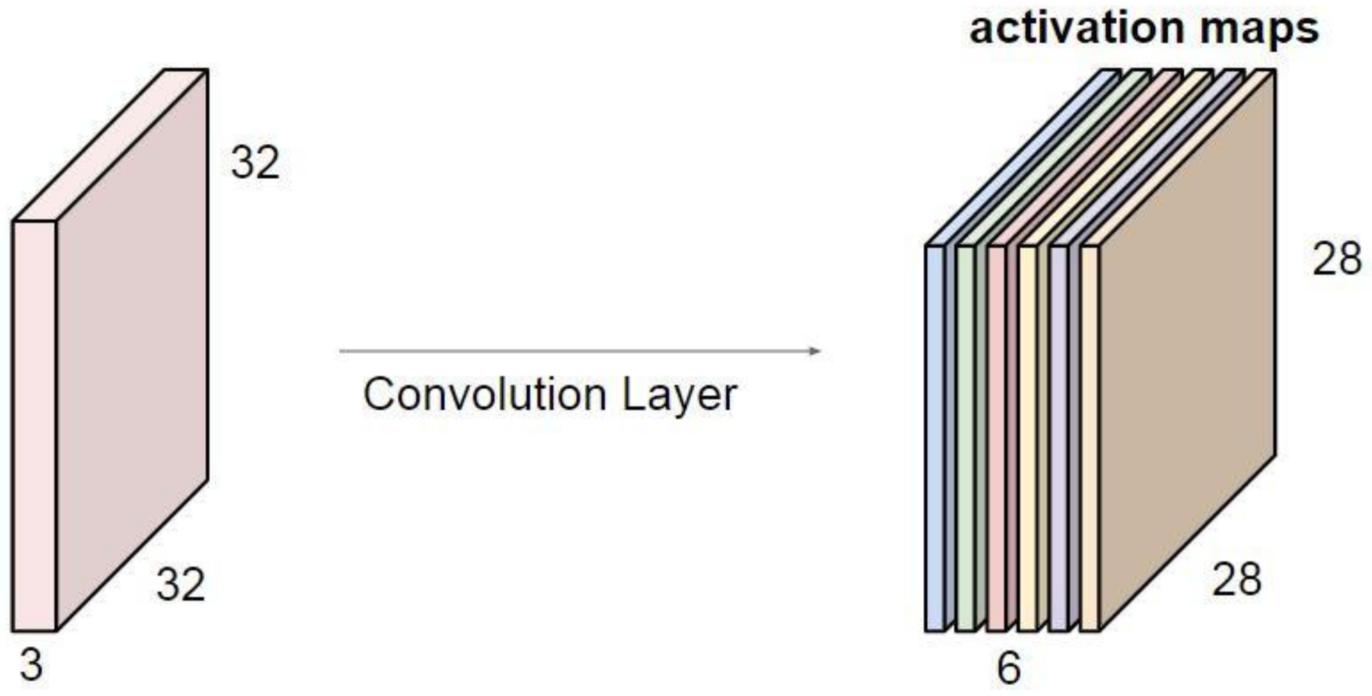
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image (i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

# Convolution Layer

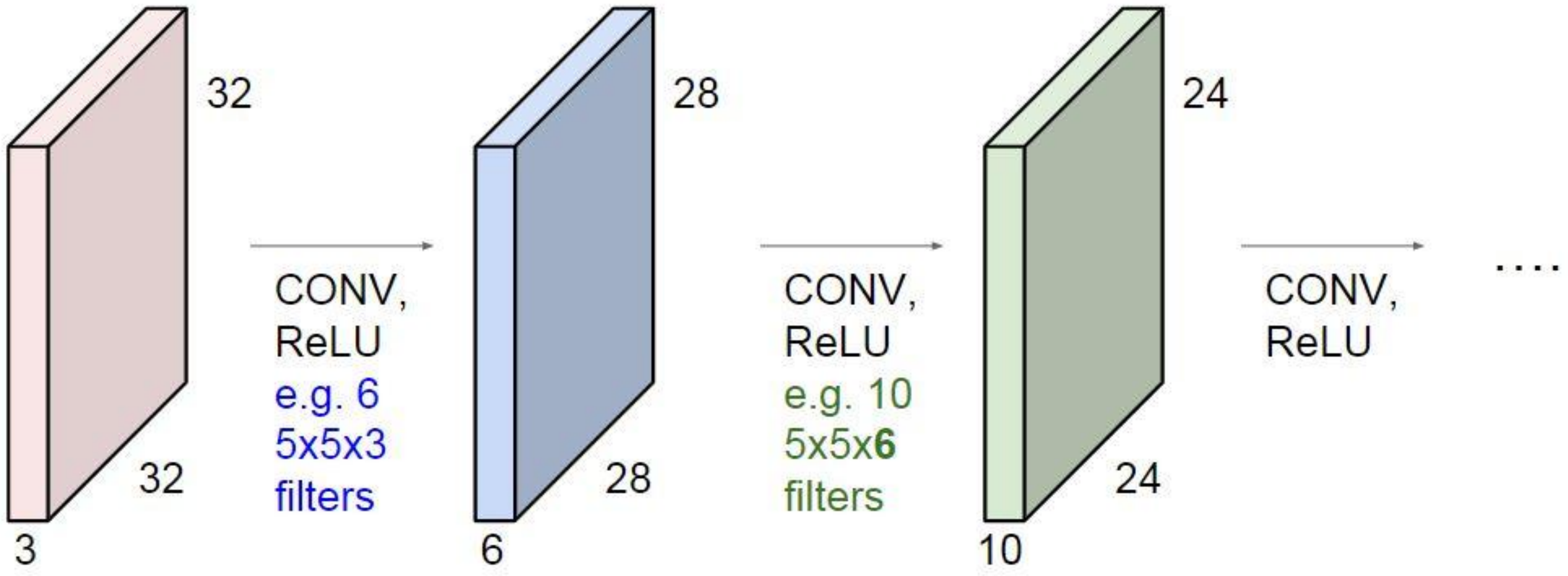


For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:

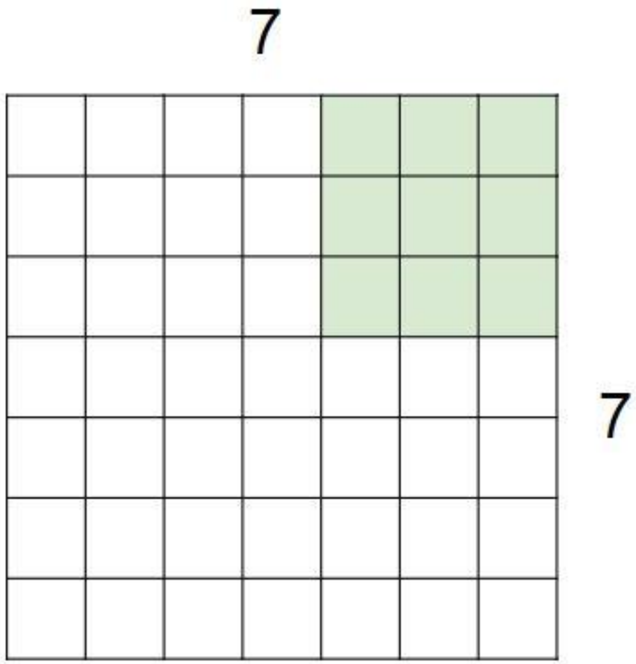



We stack these up to get a "new image" of size 28x28x6!

**Preview:** ConvNet is a sequence of Convolutional Layers, interspersed with activation functions

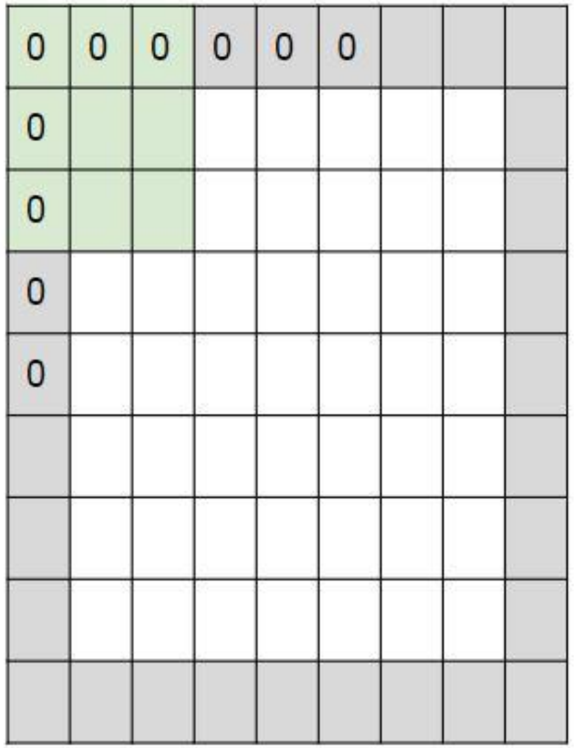


A closer look at spatial dimensions:




7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**
**=> 3x3 output!**

# In practice: Common to zero pad the border

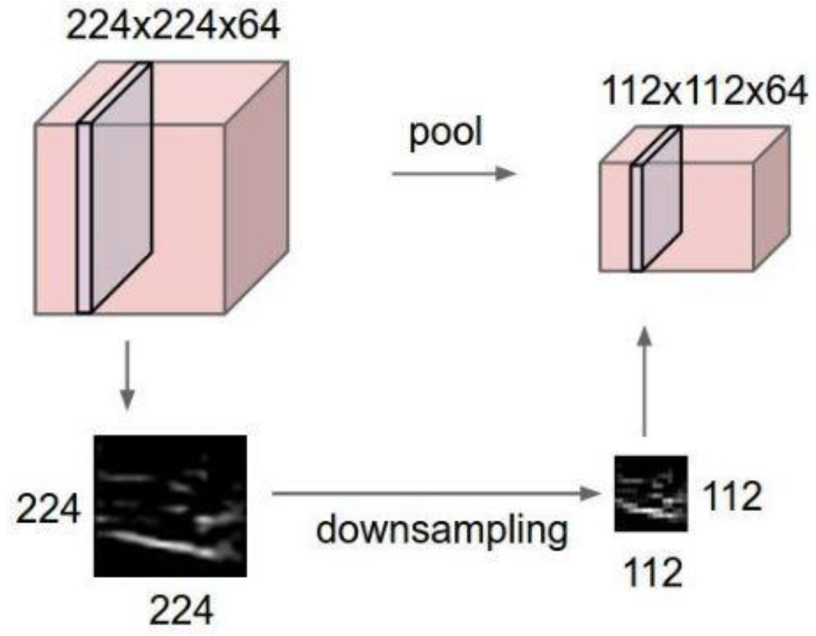



e.g. input 7x7
**3x3** filter, applied with **stride 1**
**pad with 1 pixel** border **=>** what is the output?

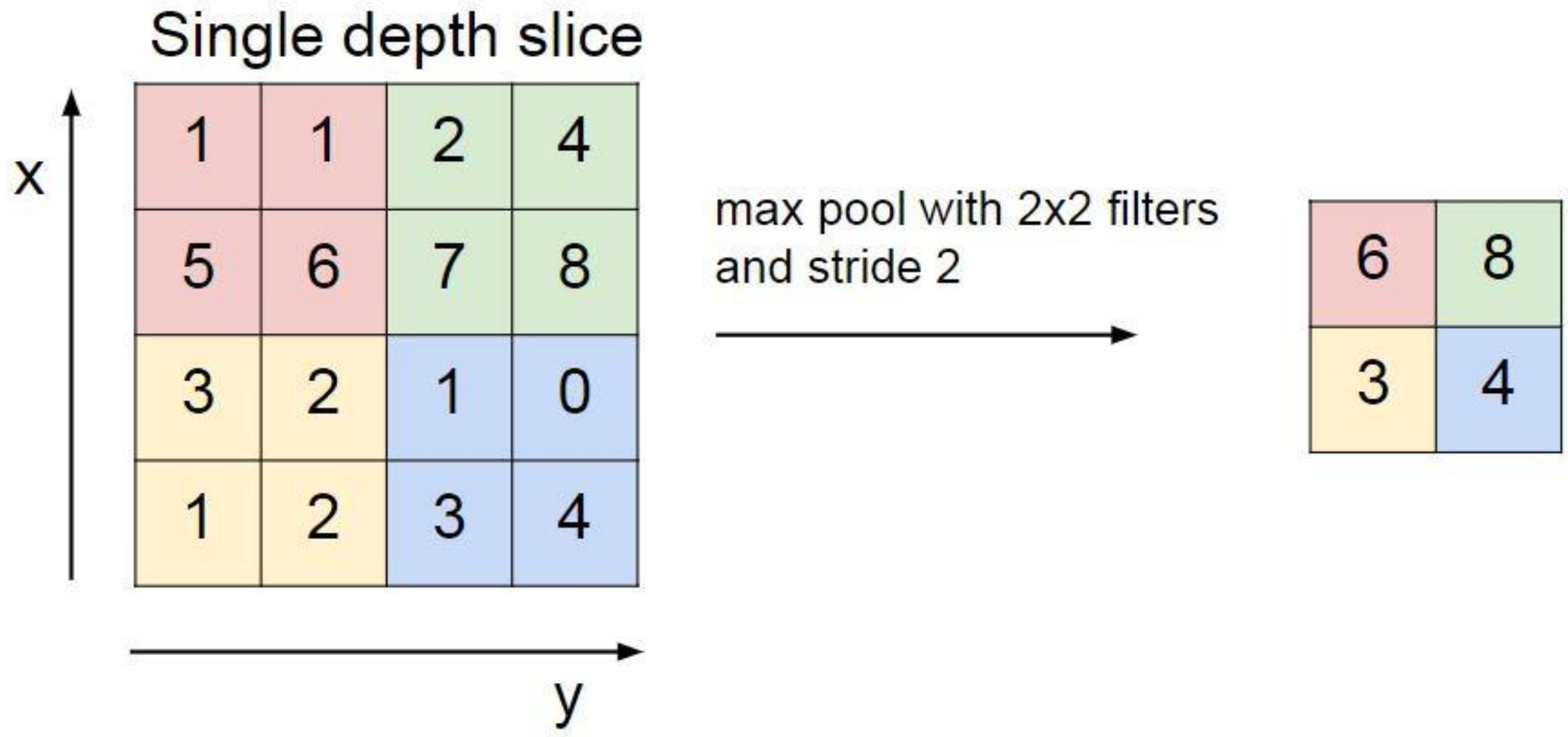
**7x7 output!**

# Pooling layer

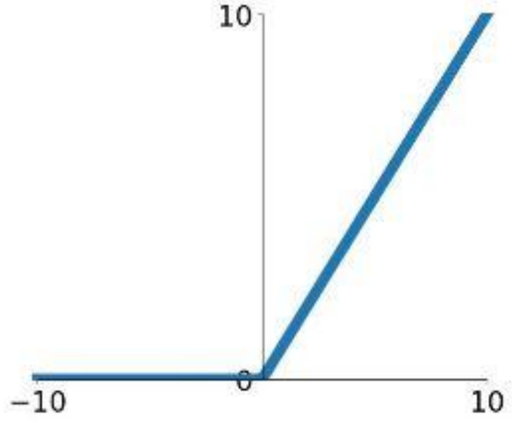
- makes the representations smaller and more manageable
- operates over each activation map independently:



# MAX POOLING


Single depth slice
x
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
y
max pool with 2x2 filters
and stride 2
6 8
3 4

# Activation Functions




**ReLU**
(Rectified Linear Unit)

- Computes **f(x) = max(0,x)**

- Does not saturate (in +region)
- Very computationally efficient
- Converges much faster than sigmoid/tanh in practice (e.g. 6x)
- Actually more biologically plausible than sigmoid

[Krizhevsky et al., 2012]

# Batch Normalization

[Ioffe and Szegedy, 2015]

"you want unit gaussian activations? just make them so."

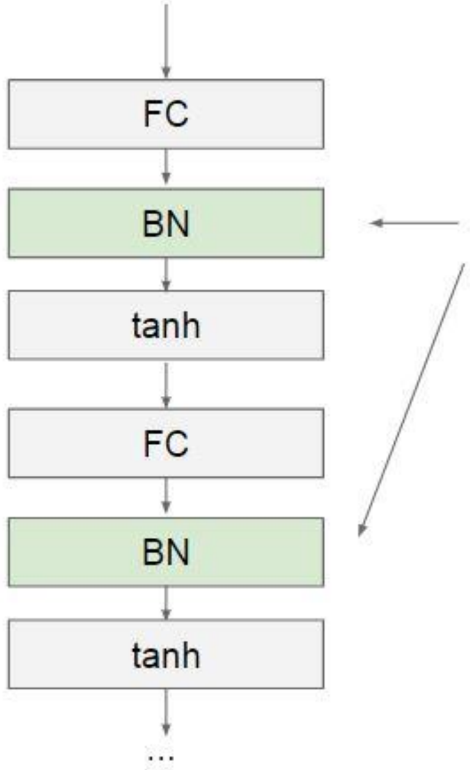
consider a batch of activations at some layer.
To make each dimension unit gaussian, apply:

$$\widehat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$

this is a vanilla differentiable function...

# Batch Normalization

[Ioffe and Szegedy, 2015]

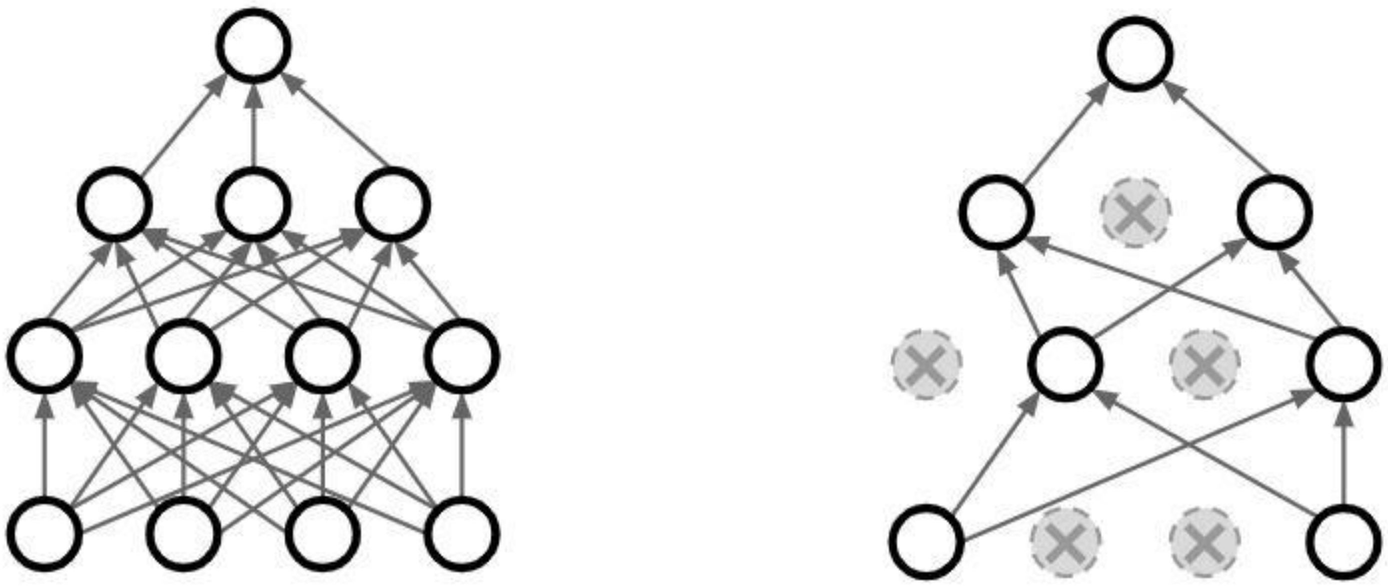



Usually inserted after Fully Connected or Convolutional layers, and before nonlinearity.

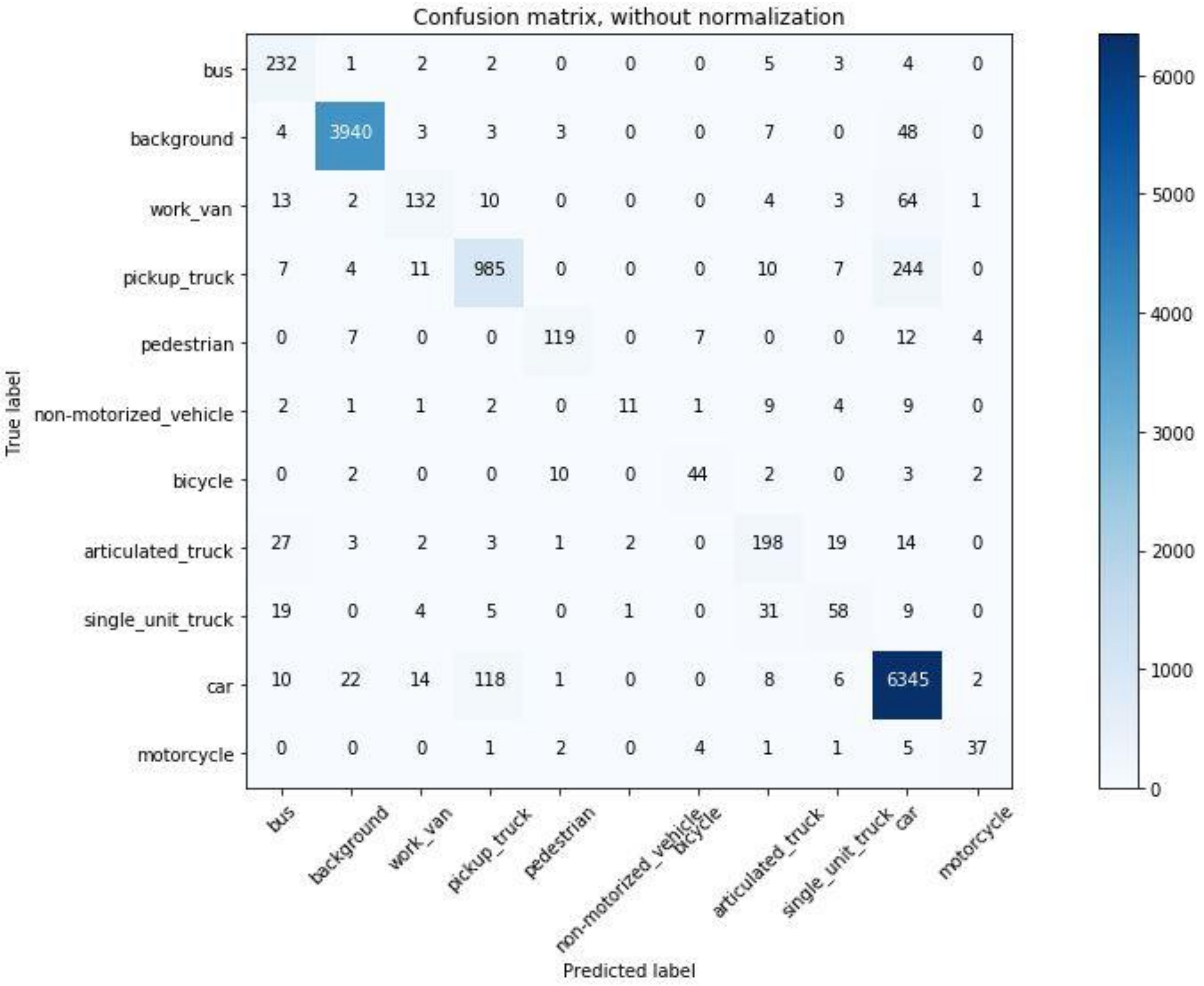
$$\widehat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$

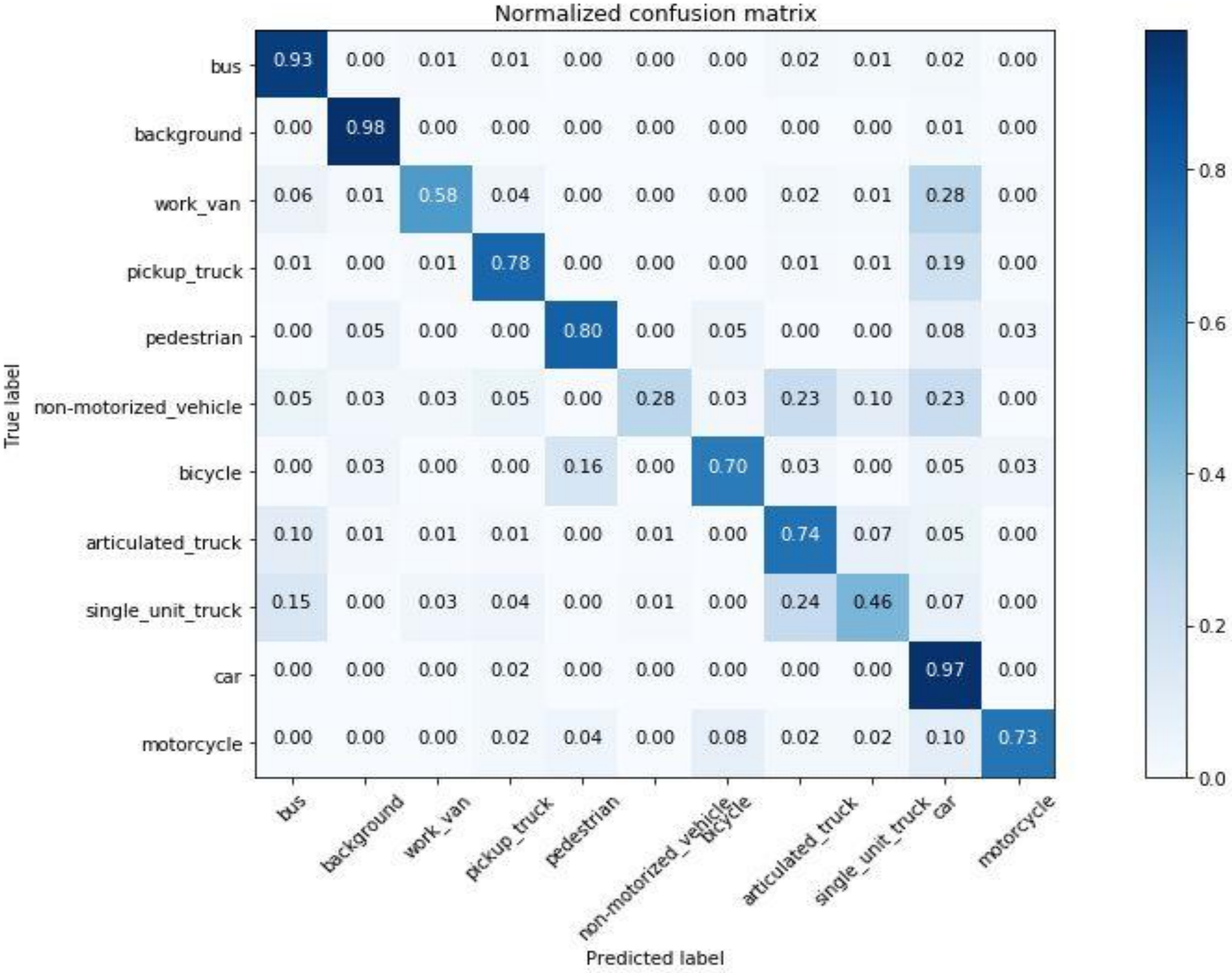
# Regularization: Dropout

In each forward pass, randomly set some neurons to zero
Probability of dropping is a hyperparameter; 0.5 is common



Srivastava et al, "Dropout: A simple way to prevent neural networks from overfitting", JMLR 2014


Confusion matrix, without normalization
True label
Predicted label
bus
background
work_van
pickup_truck
pedestrian
non-motorized_vehicle
bicycle
articulated_truck
single_unit_truck
car
motorcycle
232 1 2 2 0 0 0 5 3 4 0
4 3940 3 3 3 0 0 7 0 48 0
13 2 132 10 0 0 0 4 3 64 1
7 4 11 985 0 0 0 10 7 244 0
0 7 0 0 119 0 7 0 0 12 4
2 1 1 2 0 11 1 9 4 9 0
0 2 0 0 10 0 44 2 0 3 2
27 3 2 3 1 2 0 198 19 14 0
19 0 4 5 0 1 0 31 58 9 0
10 22 14 118 1 0 0 8 6 6345 2
0 0 0 1 2 0 4 1 1 5 37
6000
5000
4000
3000
2000
1000
0



Normalized confusion matrix
True label
Predicted label
bus
background
work_van
pickup_truck
pedestrian
non-motorized_vehicle
bicycle
articulated_truck
single_unit_truck
car
motorcycle
0.93 0.00 0.01 0.01 0.00 0.00 0.00 0.02 0.01 0.02 0.00
0.00 0.98 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.00
0.06 0.01 0.58 0.04 0.00 0.00 0.00 0.02 0.01 0.28 0.00
0.01 0.00 0.01 0.78 0.00 0.00 0.00 0.01 0.01 0.19 0.00
0.00 0.05 0.00 0.00 0.80 0.00 0.05 0.00 0.00 0.08 0.03
0.05 0.03 0.03 0.05 0.00 0.28 0.03 0.23 0.10 0.23 0.00
0.00 0.03 0.00 0.00 0.16 0.00 0.70 0.03 0.00 0.05 0.03
0.10 0.01 0.01 0.01 0.00 0.01 0.00 0.74 0.07 0.05 0.00
0.15 0.00 0.03 0.04 0.00 0.01 0.00 0.24 0.46 0.07 0.00
0.00 0.00 0.00 0.02 0.00 0.00 0.00 0.00 0.00 0.97 0.00
0.00 0.00 0.00 0.02 0.04 0.00 0.08 0.02 0.02 0.10 0.73
0.0
0.2
0.4
0.6
0.8

# References


- R.S Vaddi, L.N.P Boggavarapu, K.R Anne, "Computer Vision based Vehicle Recognition on Indian Roads" International Journal of Computer Vision and Signal Processing, 5(1), 8-13(2015)
- Chris Harris , Mike Stephens, "A combined corner and edge detector" (1988)
- D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, Int. J. Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.
- R.S Vaddi, L.N.P Boggavarapu, K.R Anne, "Indian Vehicle Database", includes four classes of testing and training images (Truck, Auto, Bus and Car resp.)
- L. Fei-Fei, R. Fergus and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. IEEE. CVPR 2004
- Z. Luo et al., "MIO-TCD: A New Benchmark Dataset for Vehicle Classification and Localization," in IEEE Transactions on Image Processing, vol. 27, no. 10, pp. 5129-5141, Oct. 2018.