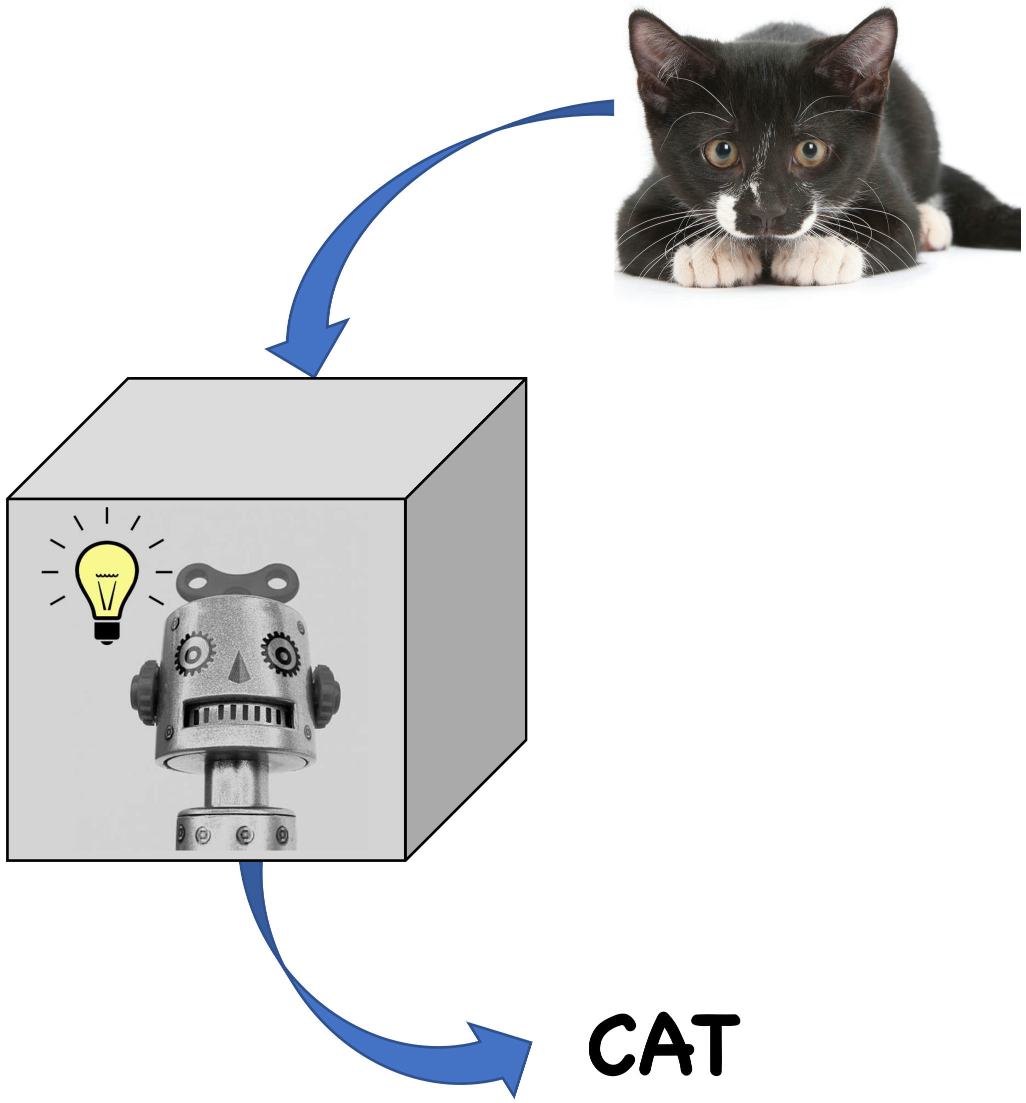
Image Classification

Susmita Datta



Dataset

CIFAR-10

10 classes

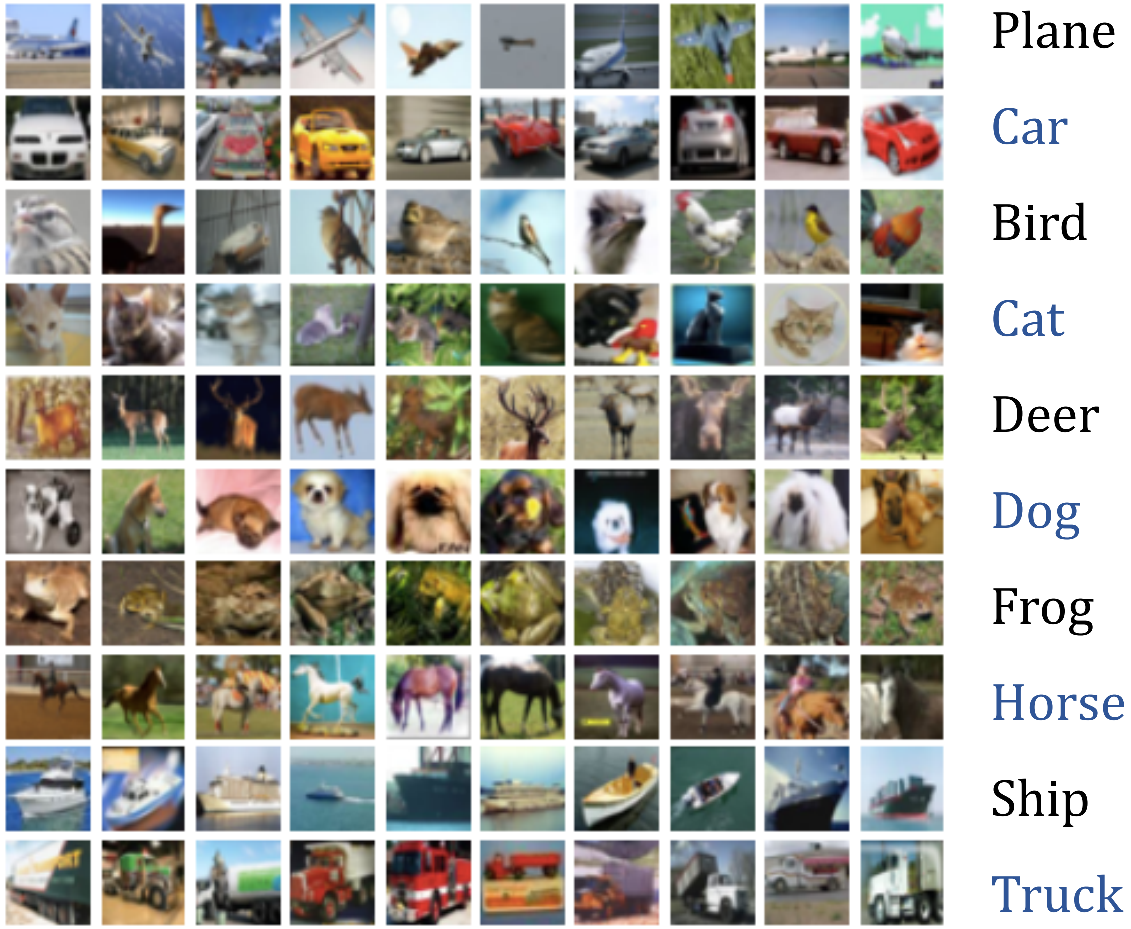
60,000 RGBs

- 50,000 training
- 10,000 test

32 pix × 32 pix × 3 pix

Data source

https://www.cs.toronto.edu/~kriz/cifar.html



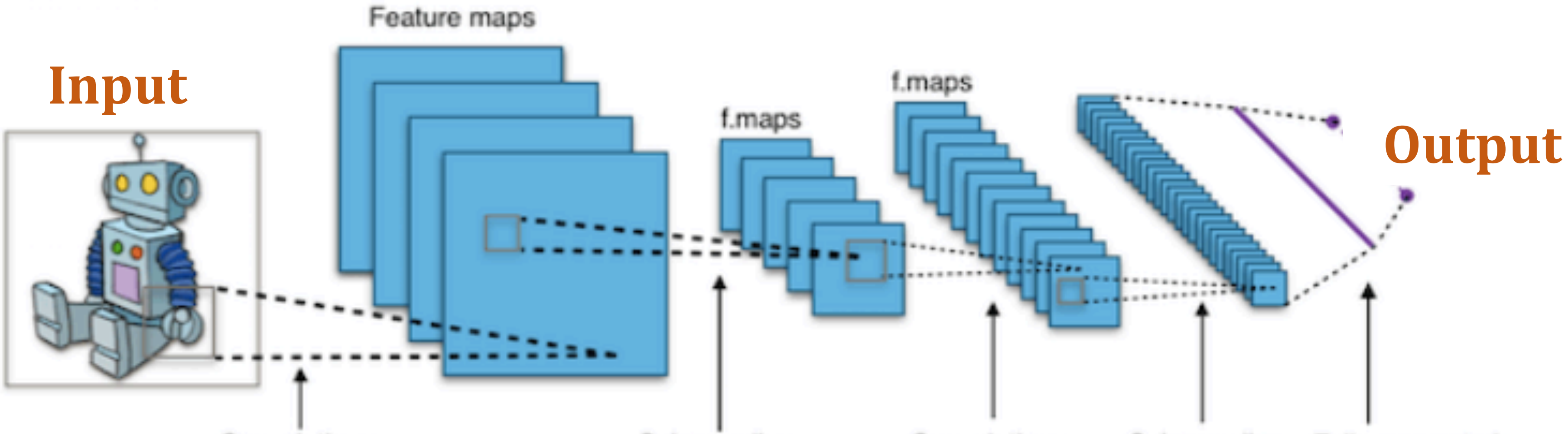
Classification Models

	Accuracy
Convolutional Neural Network	**83%**
K-nearest Neighbors	27%
Support Vector Machine	47%

Convolutional Neural Network

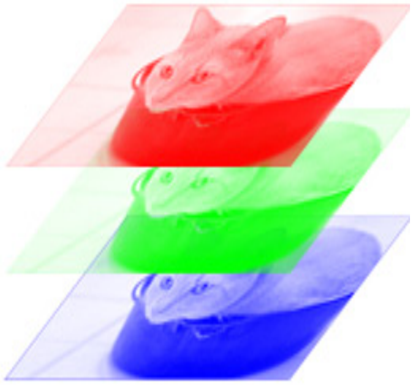
Convolutional Neural Network (ConvNet or CNN)

A deep learning process explicitly for images



Input goes through a series of transformations

ConvNet Sequence Modeling



Images in batches

Combination of Layers and Transformations

Convolution

Activation

Max Pooling

Dropout

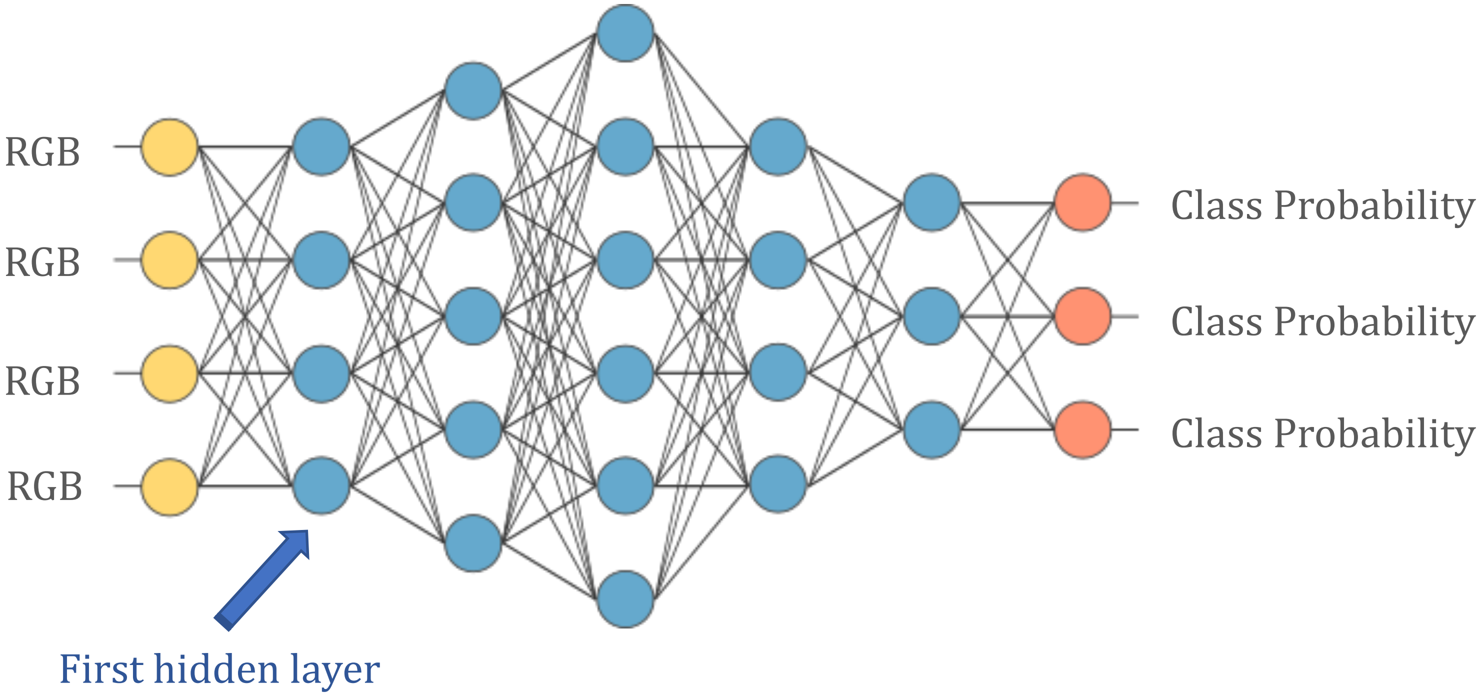
Fully Connected

1	Plane
2	Car
3	Bird
4	Cat
5	Deer
6	Dog
7	Frog
8	Horse
9	Ship
10	Truck

Class Probability

Convolution Layer

Input	Transformation	Output
RGB image	Convolve images with learnable filters	Filter response (hidden layer)



Convolution Layer

Input	Transformation	Output
RGB Images	Convolve images with learnable kernels	Kernel response

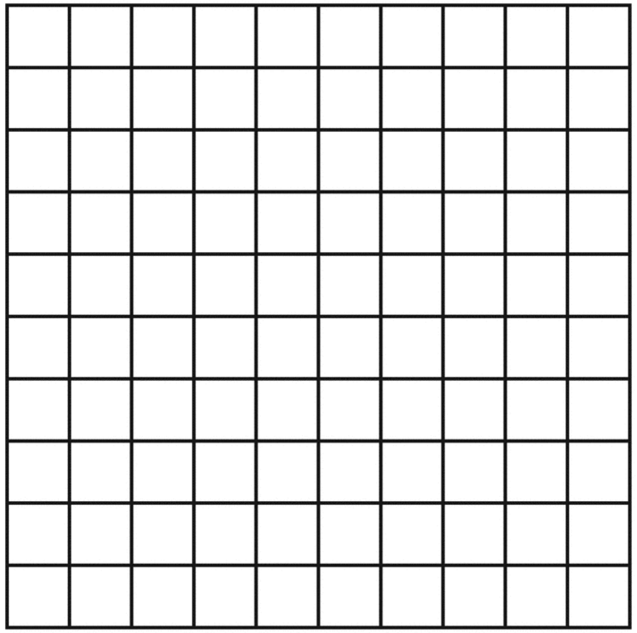
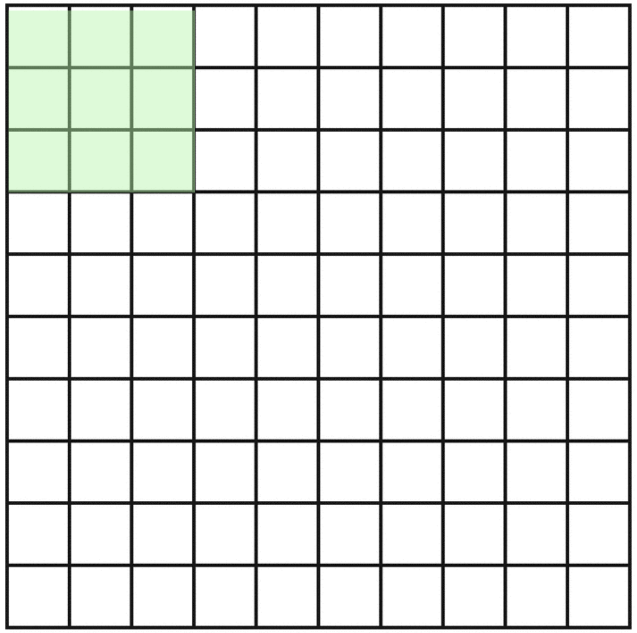


Image Pixels

Convolution applies weights to pixels

Convolution Layer

Input	Transformation	Output
RGB Images	Convolve images with learnable kernels	Kernel response



A 3 x 3 convolution kernel

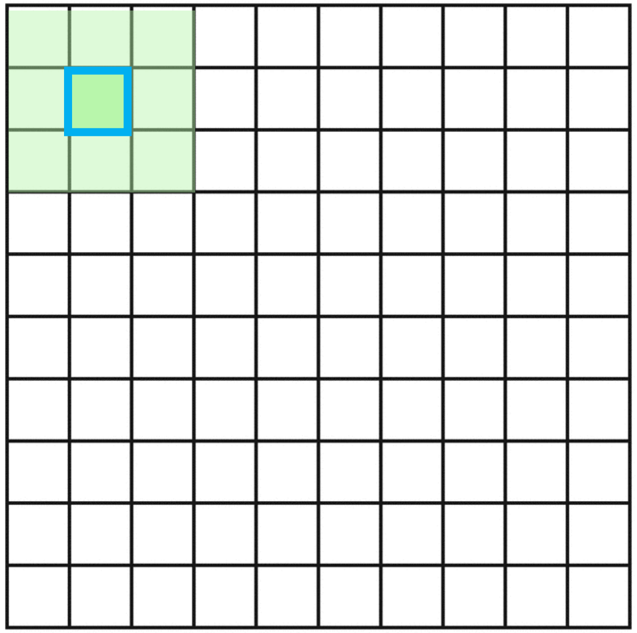
Image Pixels

Convolution applies weights to pixels

Convolution kernel is a weight matrix

Convolution Layer

Input	Transformation	Output
RGB Images	Convolve images with learnable kernels	Kernel response

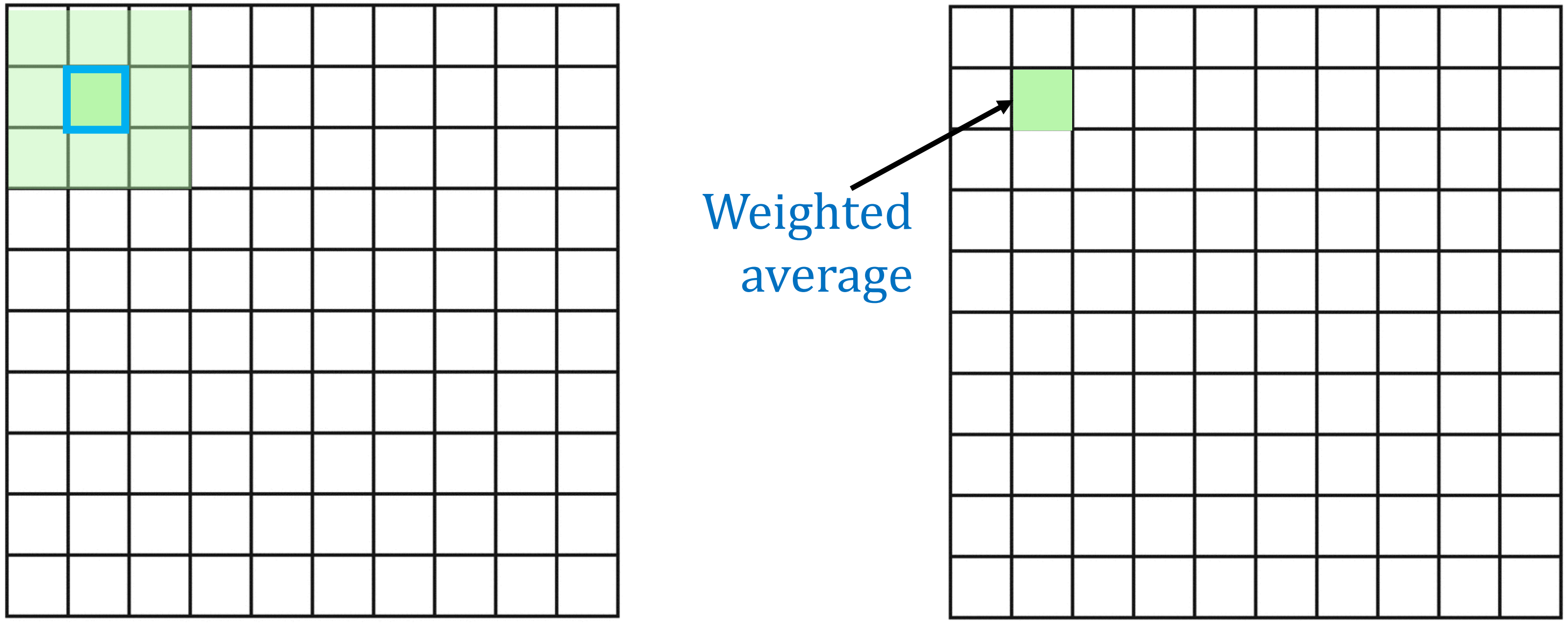


Kernel is a weight matrix

Image Pixels

Convolution Layer

Input	Transformation	Output
RGB Images	Convolve images with learnable kernels	Kernel response



Kernel is a weight matrix

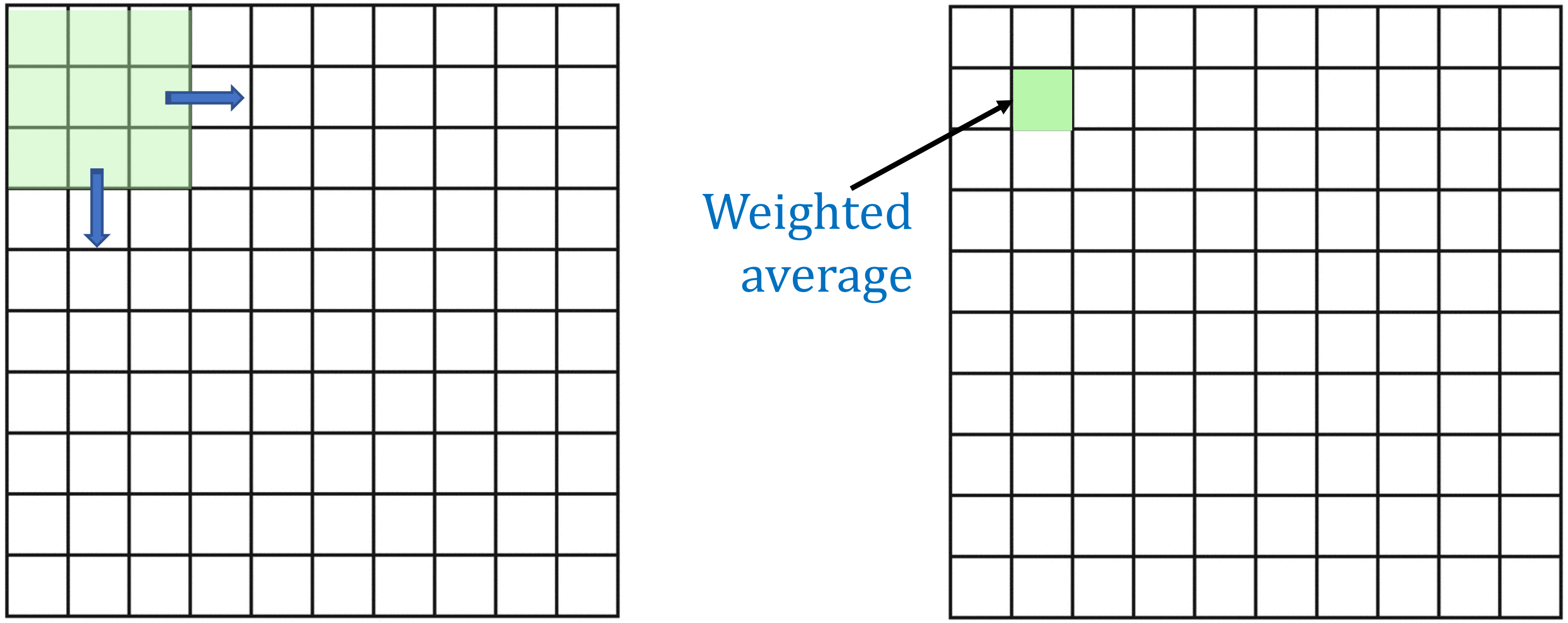
Weighted average

Image Pixels

Convolution Response

Convolution Layer

Input	Transformation	Output
RGB Images	Convolve images with learnable kernels	Kernel response



Kernel is a weight matrix

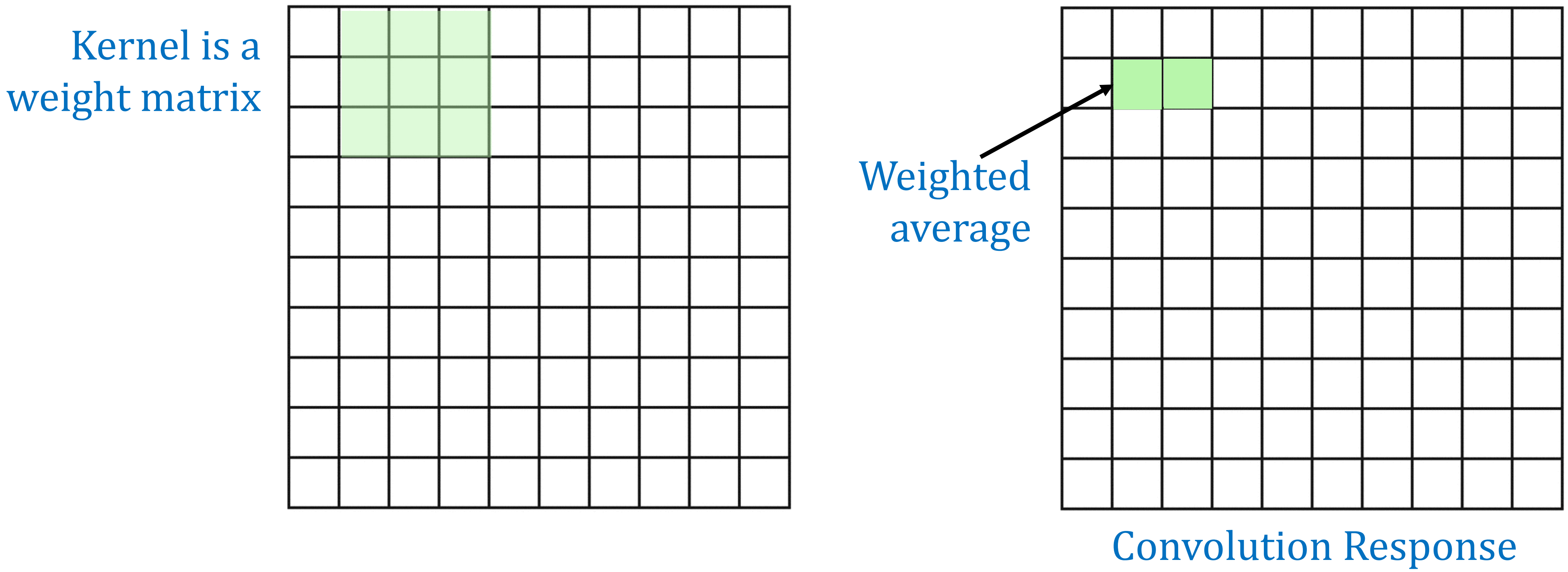
Weighted average

Image Pixels

Convolution Response

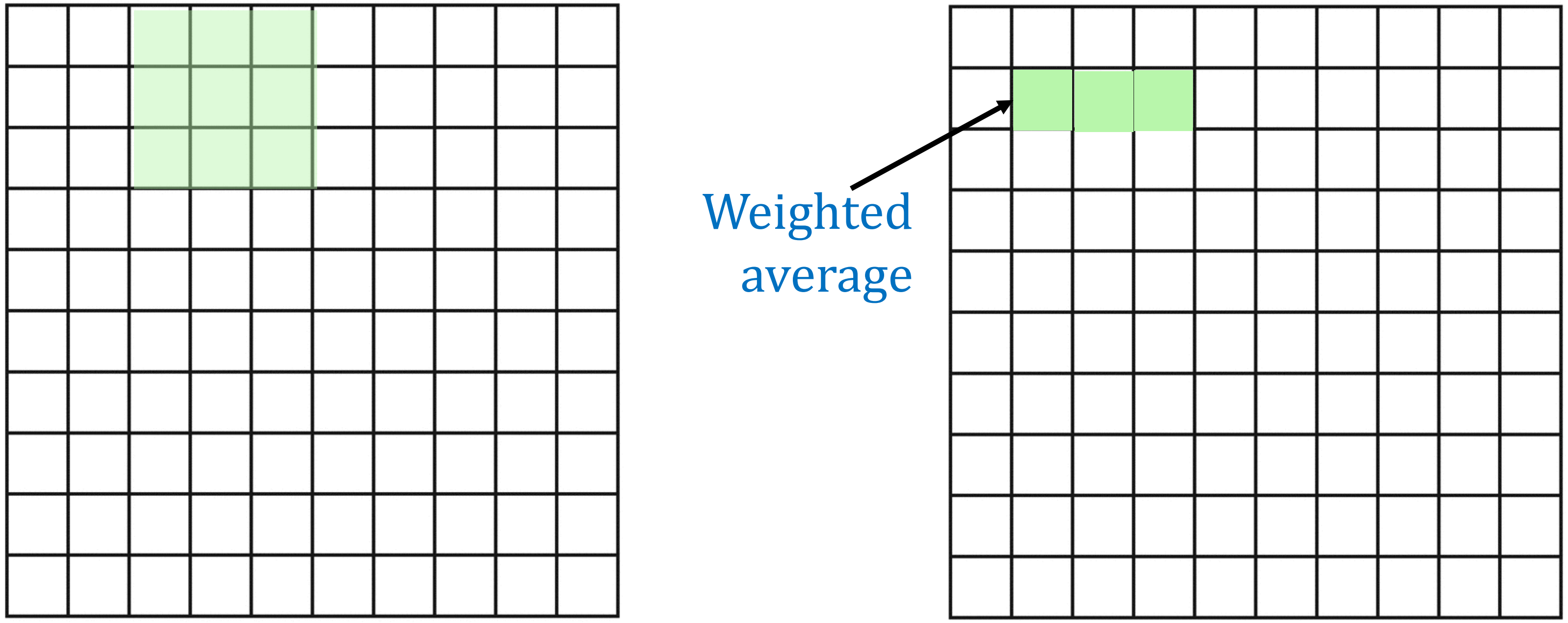
Convolution Layer

Input	Transformation	Output
RGB Images	Convolve images with learnable kernels	Kernel response



Convolution Layer

Input	Transformation	Output
RGB Images	Convolve images with learnable kernels	Kernel response



Kernel is a weight matrix

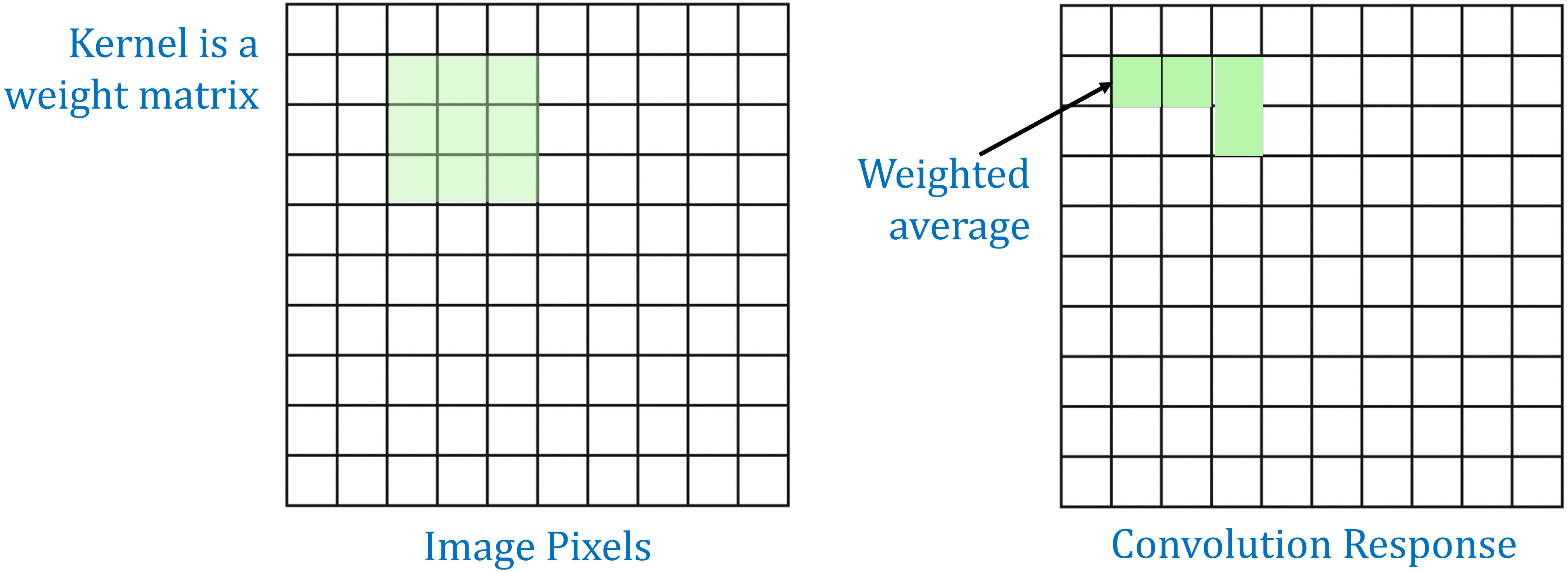
Weighted average

Image Pixels

Convolution Response

Convolution Layer

Input	Transformation	Output
RGB Images	Convolve images with learnable kernels	Kernel response



Convolution Layer

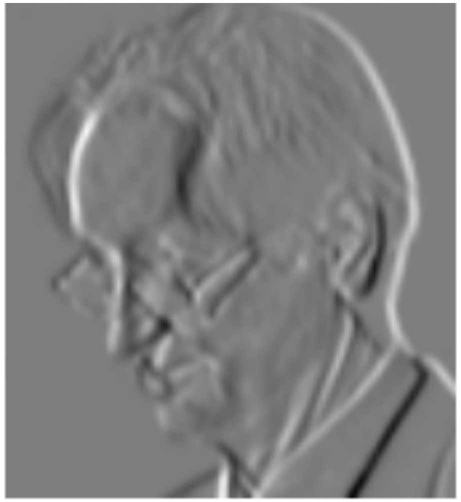
Input	Transformation	Output
RGB Images	Convolve images with learnable kernels	Kernel response

Original

*

1	0	-1
2	0	-2
1	0	-1

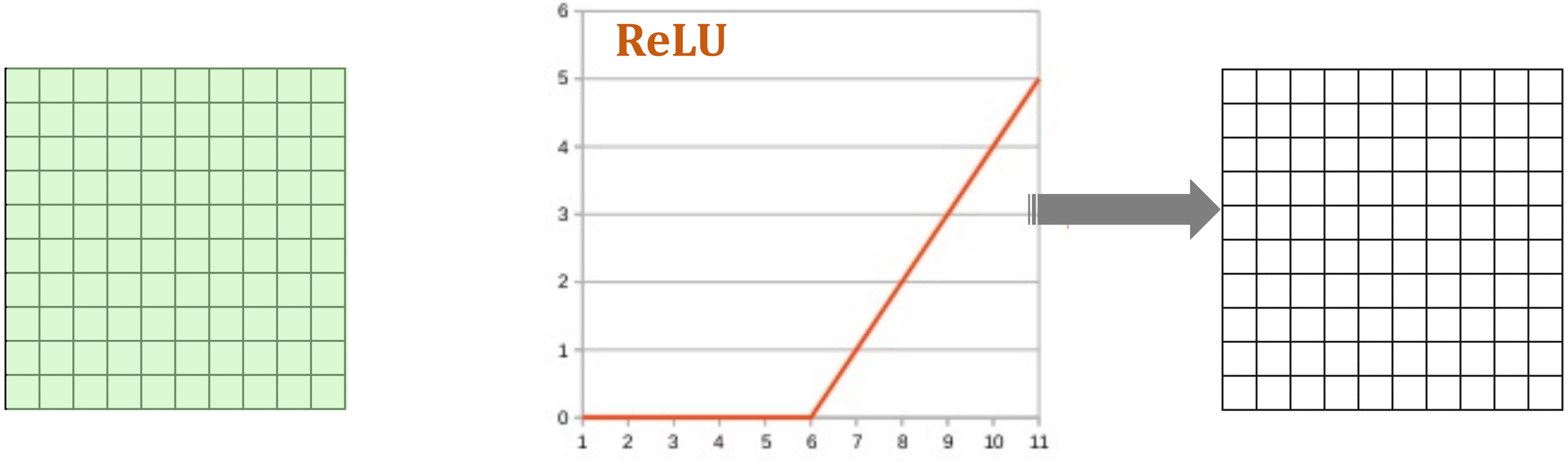
Convolution kernel



Convoluted

Activation Layer

Input	Transformation	Output
Output of Convolutional Layer	Apply ReLU function elementwise	Activation Response

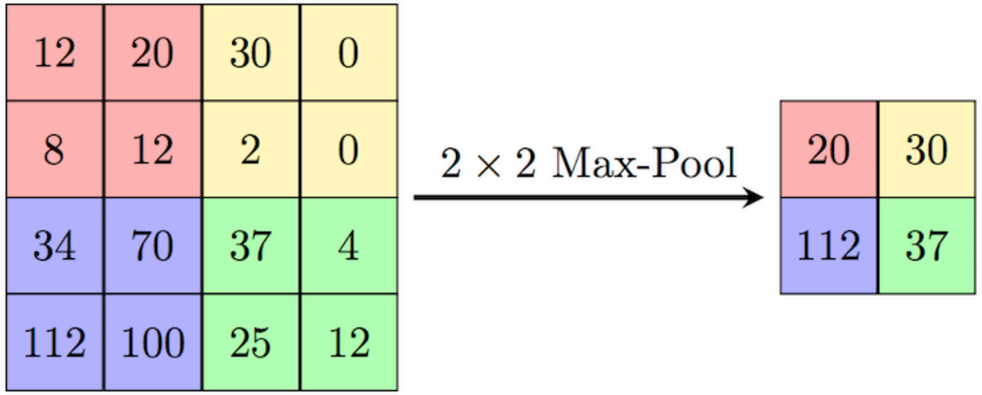


Apply a threshold

Max Pooling Layer

Input	Transformation	Output
Output of Previous Layer	Apply Max Pooling	Max-pooling image

Keep the maximum value of a region



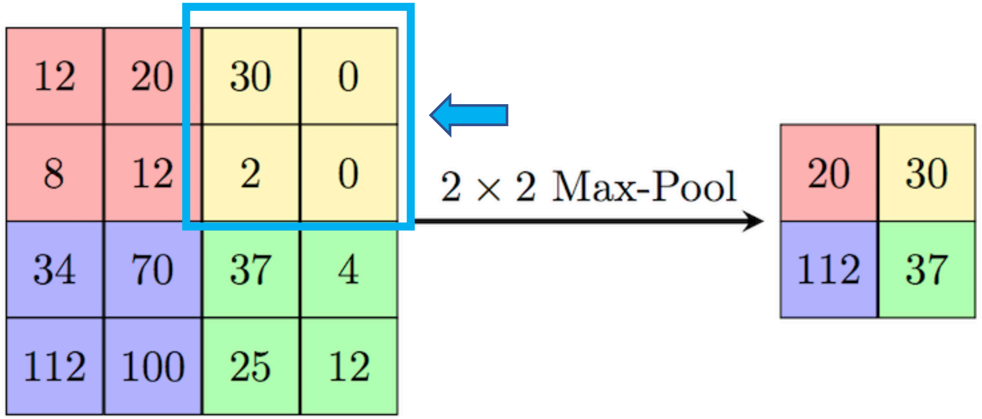
12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

20	30
112	37

Max Pooling Layer

Input	Transformation	Output
Output of Previous Layer	Apply Max Pooling	Max pooling response

Keep the maximum value of a region



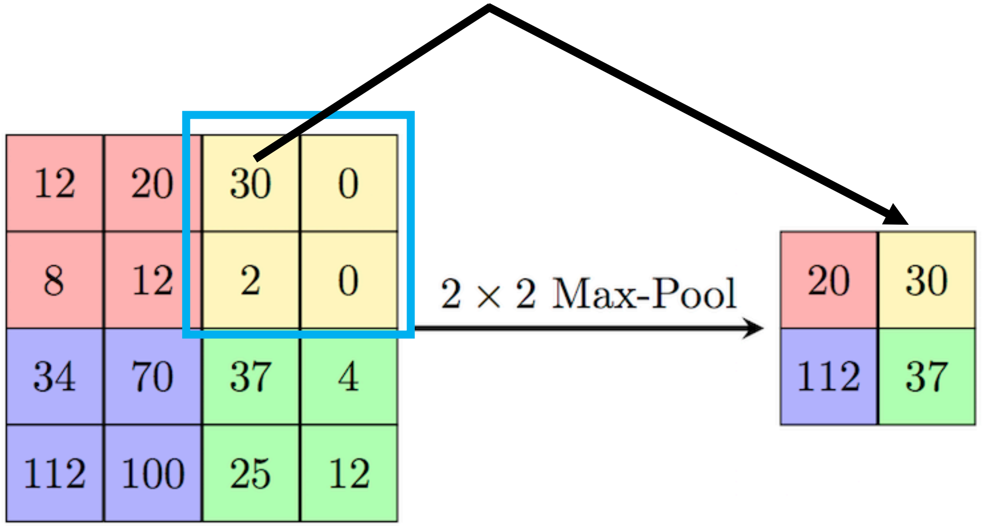
12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

20	30
112	37

Max Pooling Layer

Input	Transformation	Output
Output of Previous Layer	Apply Max Pooling	Max pooling response

Keep the maximum value of a region



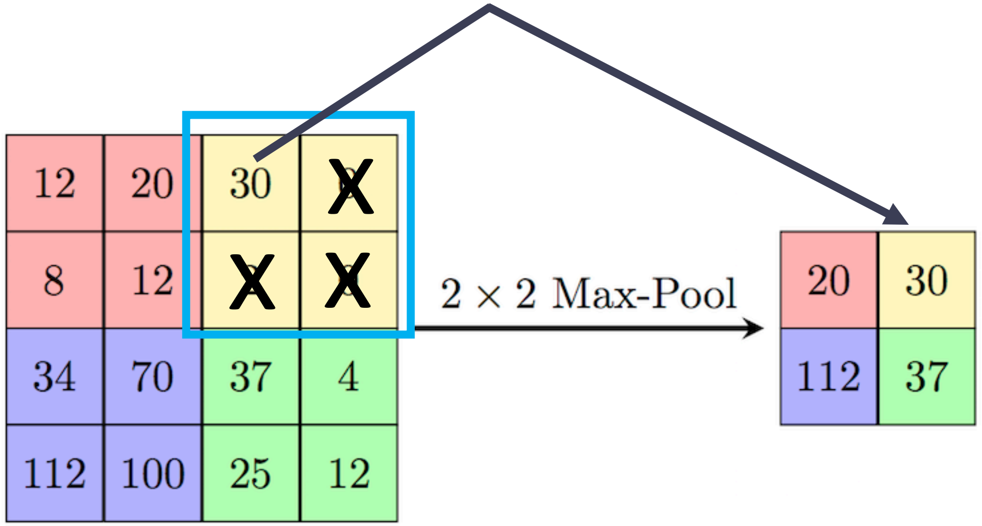
12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

20	30
112	37

Max Pooling Layer

Input	Transformation	Output
Output of Previous Layer	Apply Max Pooling	Max pooling response

Keep the maximum value of a region

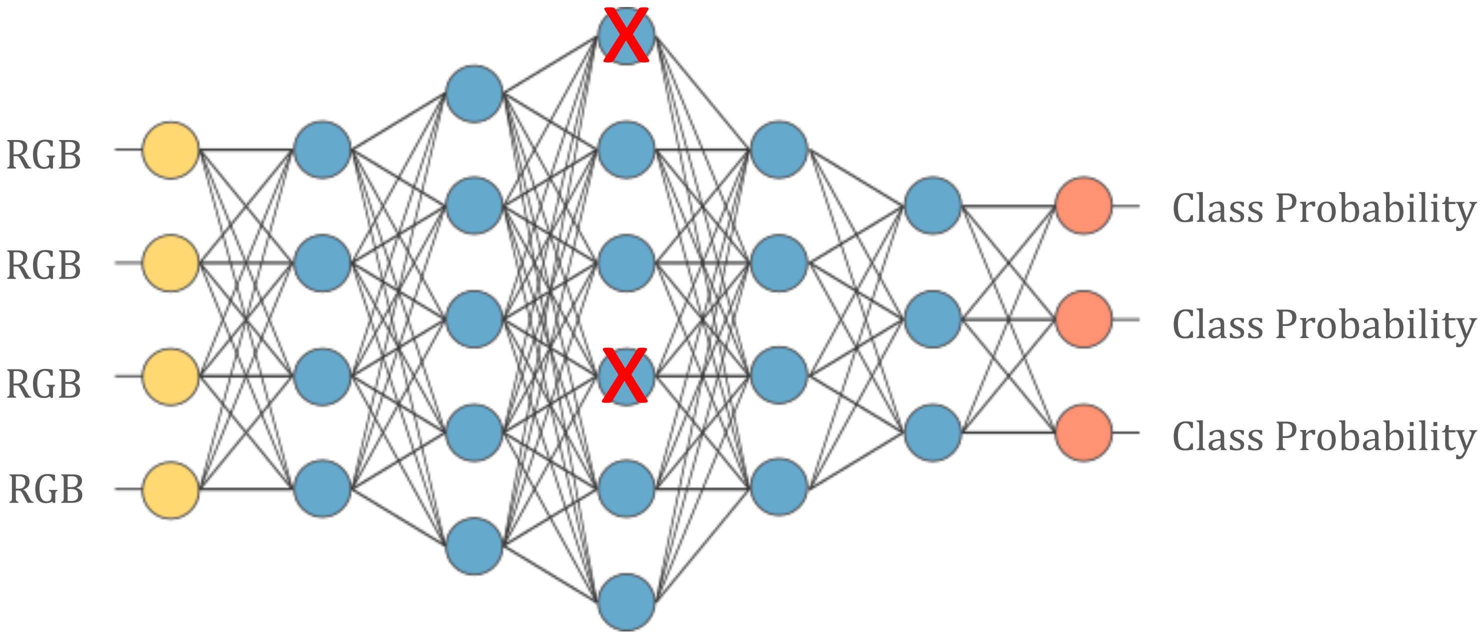


12	20	30	X
8	12	X	X
34	70	37	4
112	100	25	12

20	30
112	37

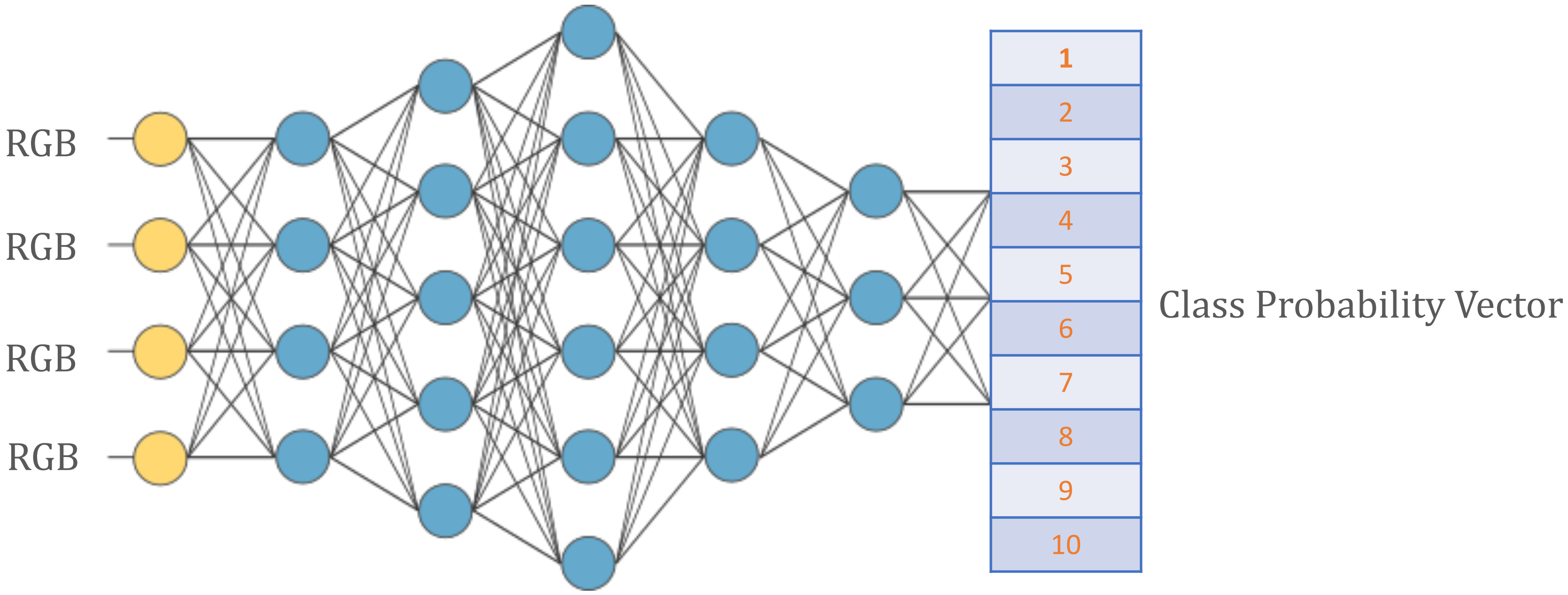
Dropout Layer

Randomly drop out neurons from a hidden layer

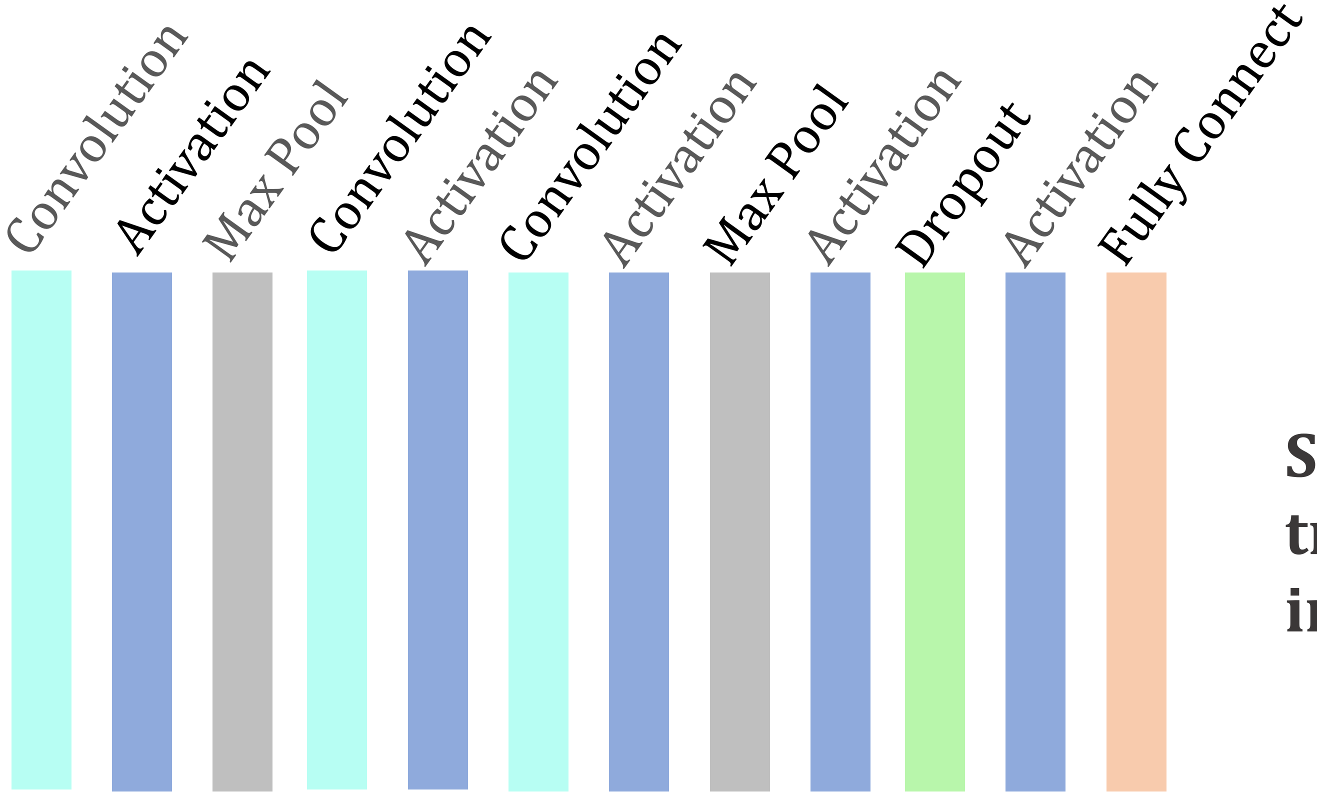


Fully Connected Layer

Input	Transformation	Output
Output from the preceding layer	Mapping Matrix multiplication	N-dim probability vector (N = number of classes)

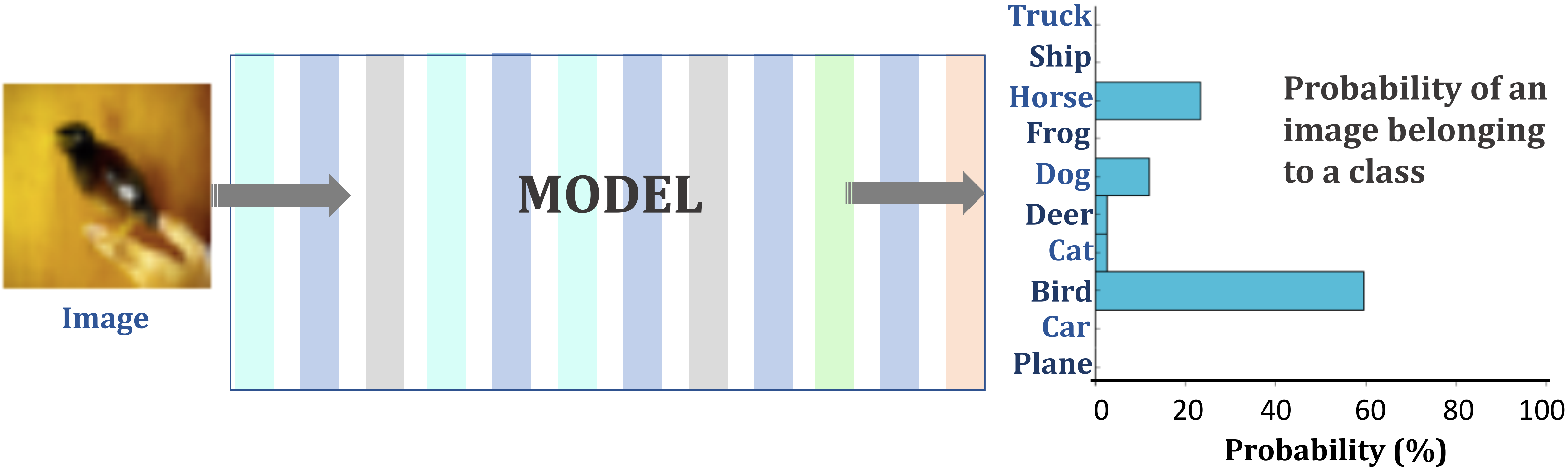


ConvNet Model Summary



Sequence of transformations in the model

ConvNet Model Summary

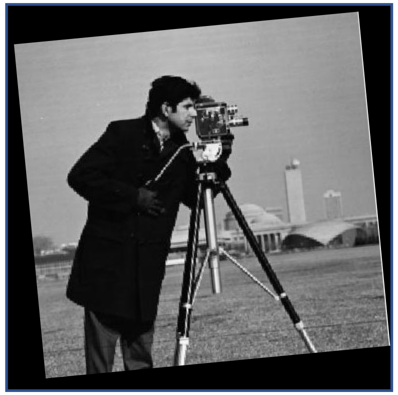


Probability of an image belonging to a class

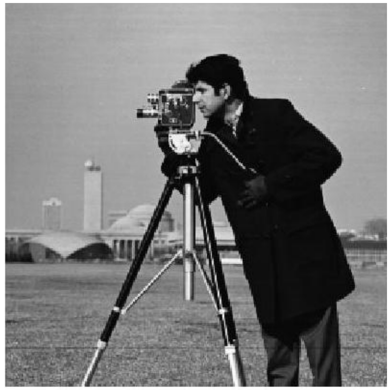
Training Set

Image Augmentation

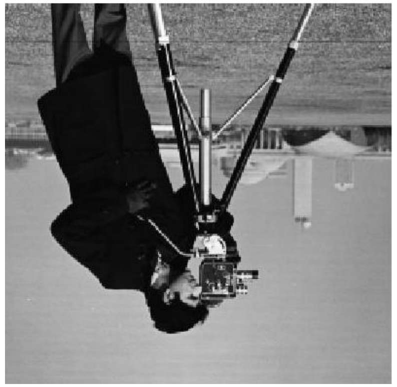
Original



Rotation



Horizontal Flips

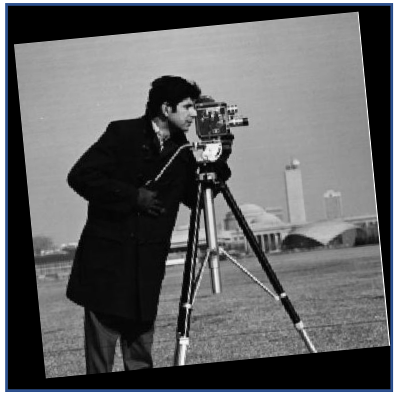


Vertical Flips

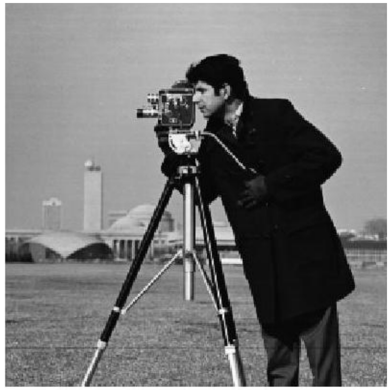
Increasing the Training Set

Image Augmentation

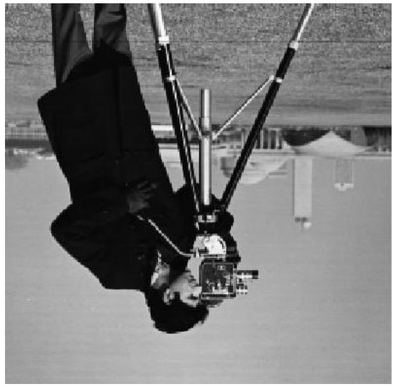
Original



Rotation



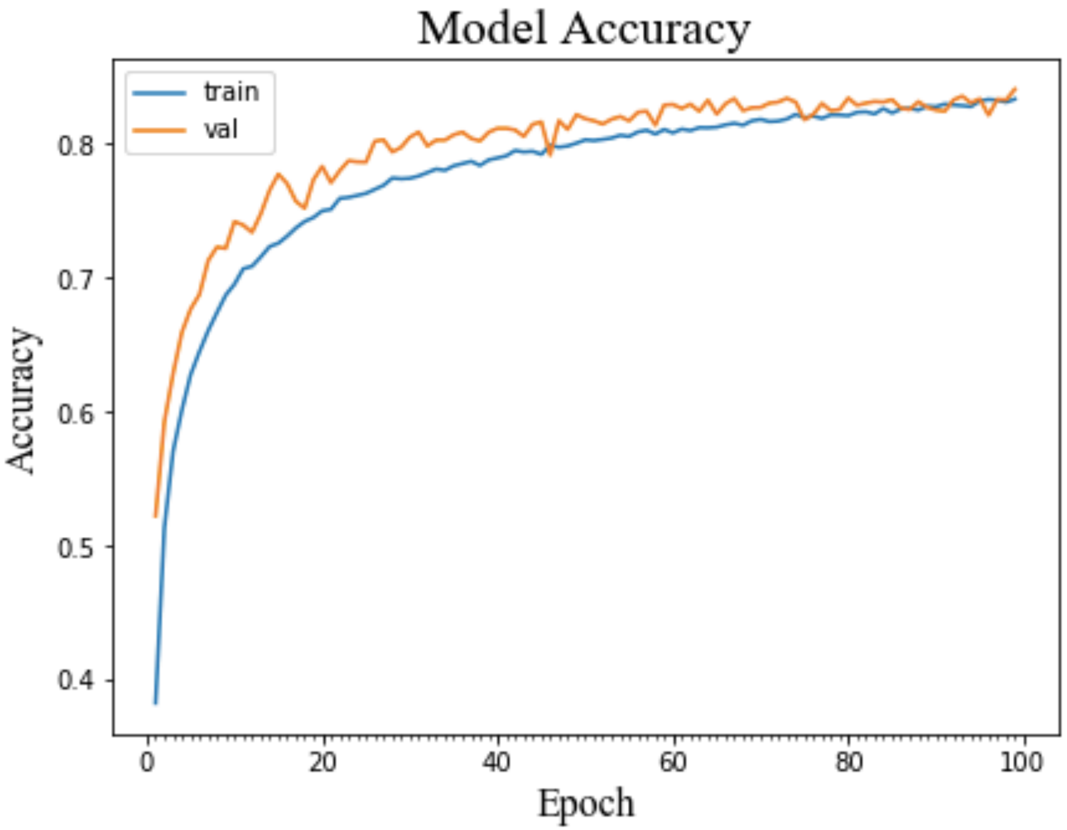
Horizontal Flips

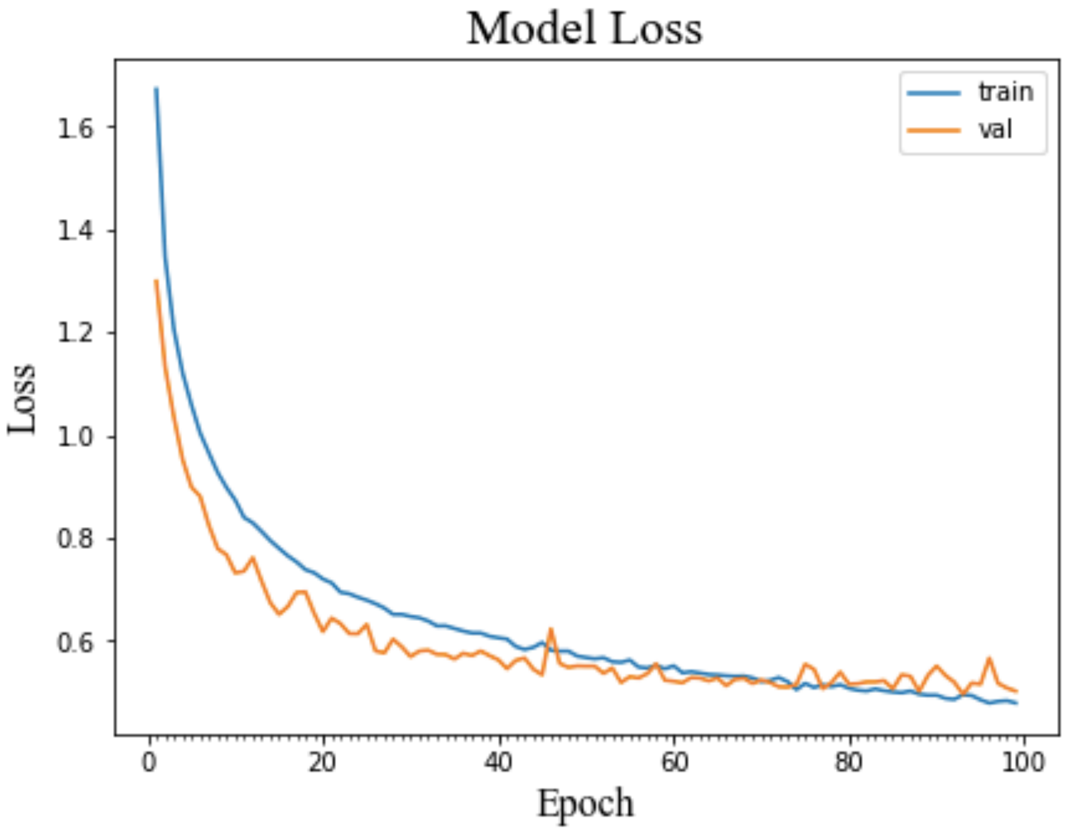


Vertical Flips

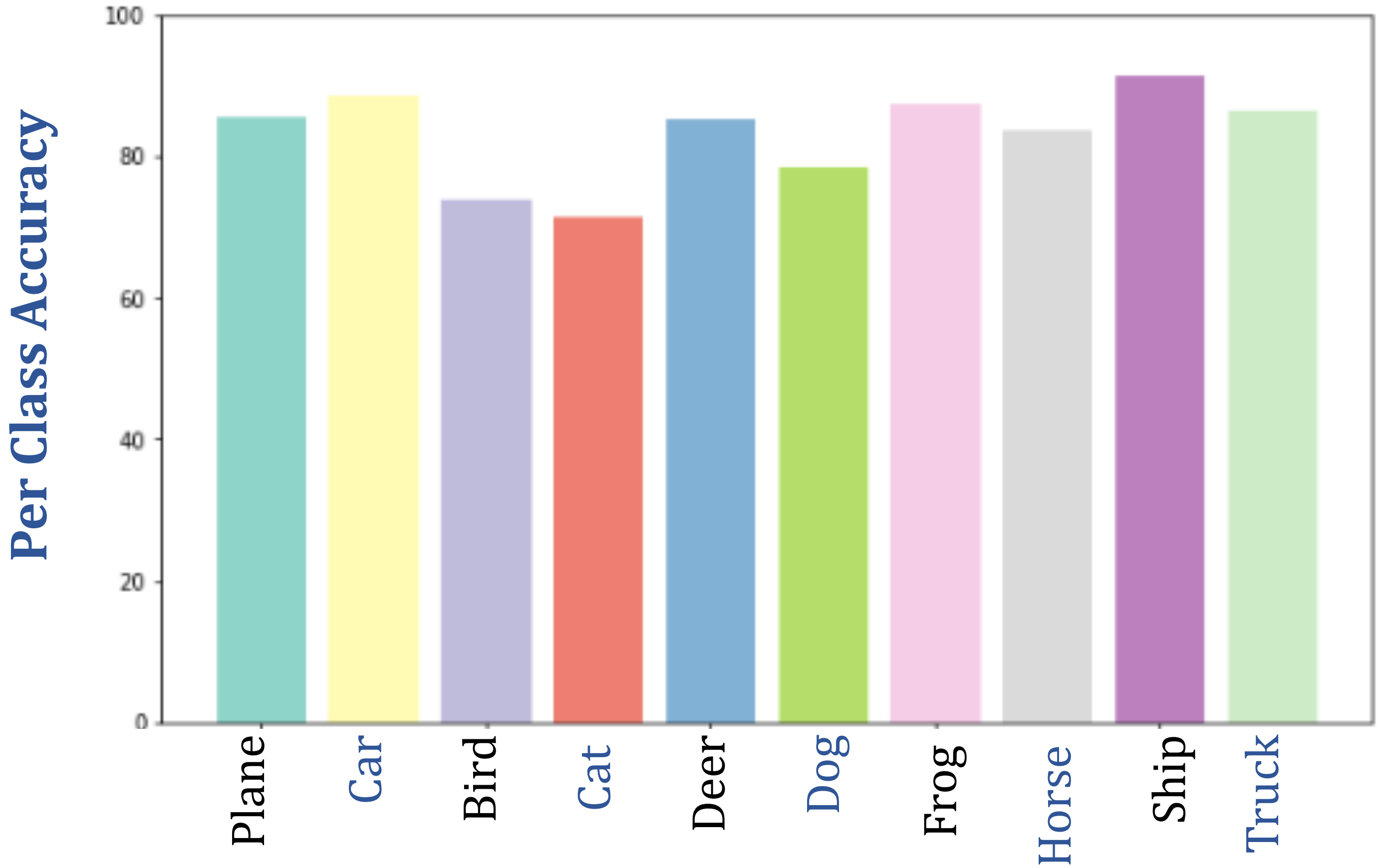
Larger training set = original training set + augmented images

Accuracy and Loss



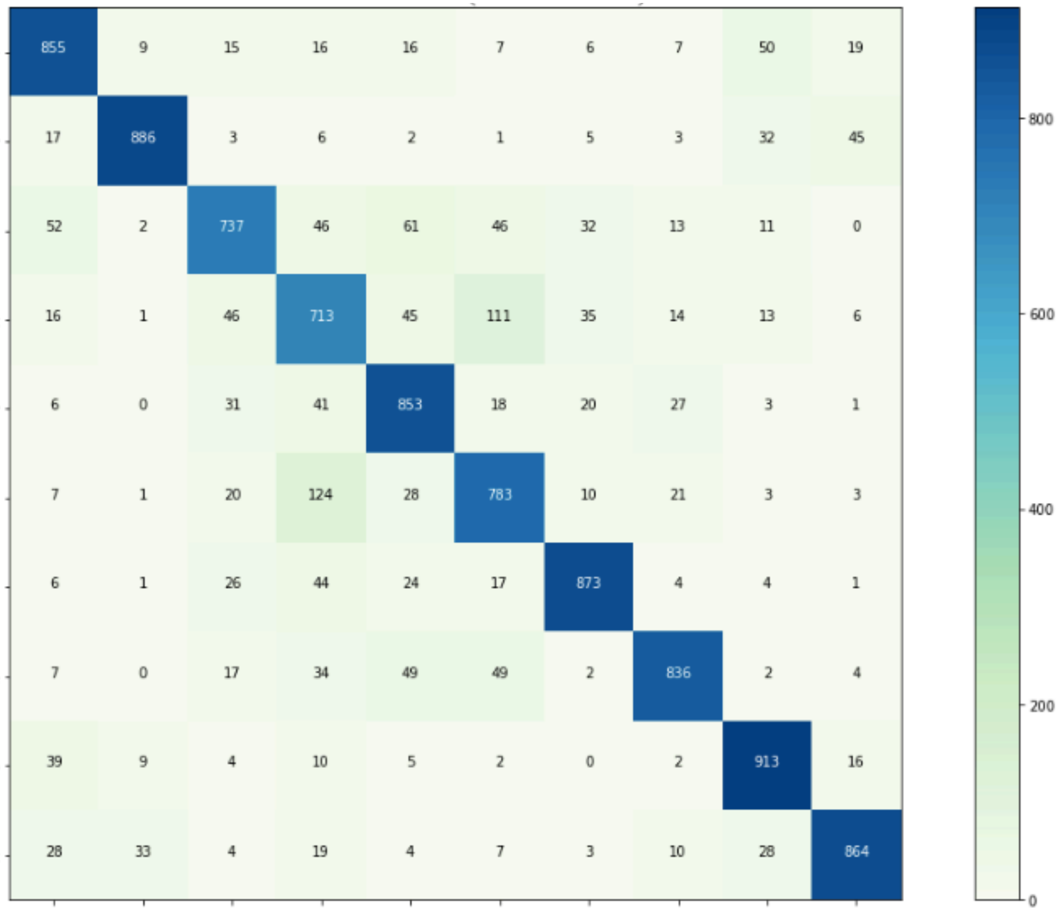


Classification Accuracy



Overall accuracy: 83%

Confusion Matrix

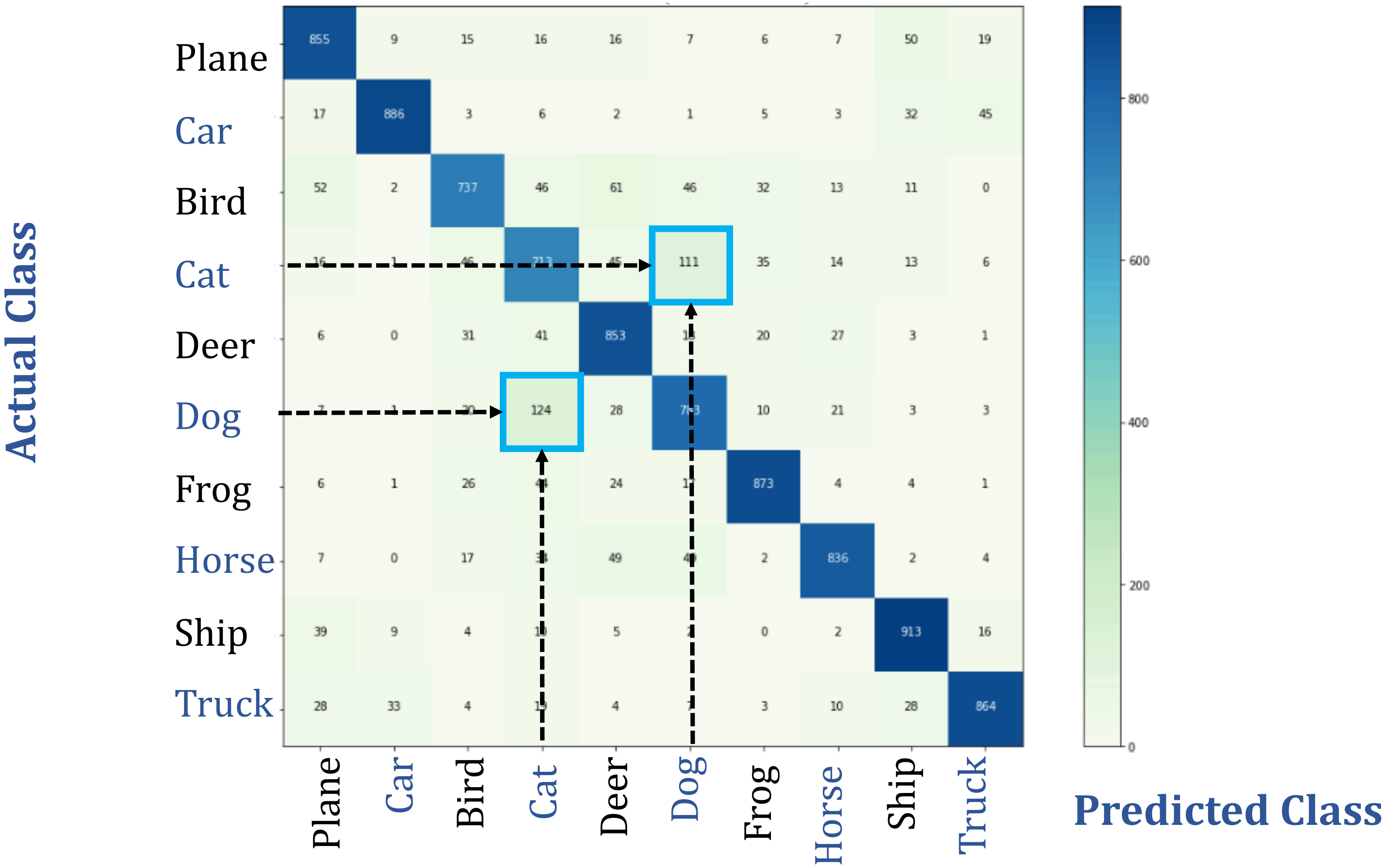


Actual Class

	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
Plane	855	9	15	16	16	7	6	7	50	19
Car	17	886	3	6	2	1	5	3	32	45
Bird	52	2	737	46	61	46	32	13	11	0
Cat	16	1	46	713	45	111	35	14	13	6
Deer	6	0	31	41	853	18	20	27	3	1
Dog	7	1	20	124	28	783	10	21	3	3
Frog	6	1	26	44	24	17	873	4	4	1
Horse	7	0	17	34	49	49	2	836	2	4
Ship	39	9	4	10	5	2	0	2	913	16
Truck	28	33	4	19	4	7	3	10	28	864

Predicted Class

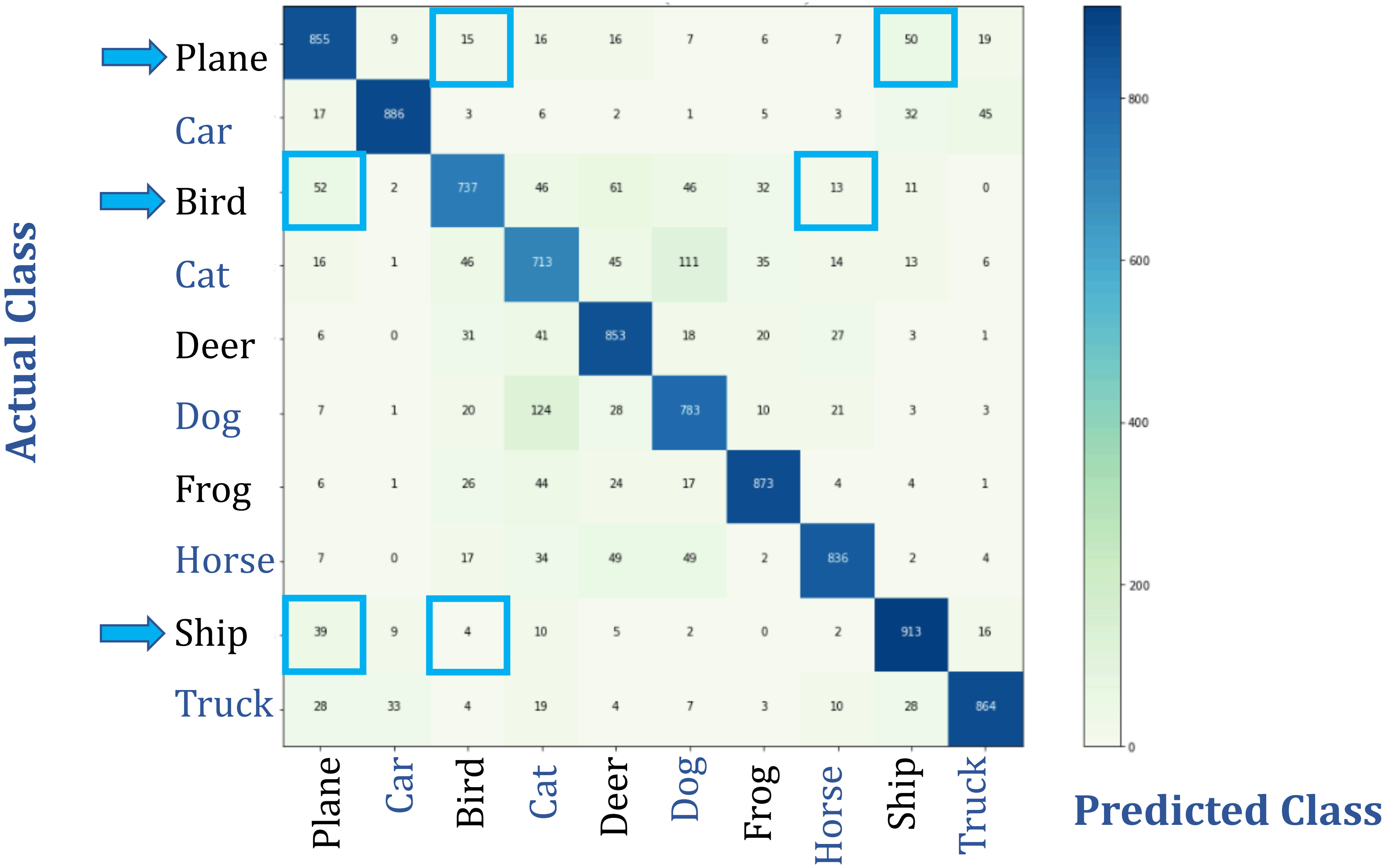
Confusion Matrix



Actual Class	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
Plane	855	9	15	16	16	7	6	7	50	19
Car	17	886	3	6	2	1	5	3	32	45
Bird	52	2	737	46	61	46	32	13	11	0
Cat	16	1	46	713	45	111	35	14	13	6
Deer	6	0	31	41	853	18	20	27	3	1
Dog	7	1	20	124	28	783	10	21	3	3
Frog	6	1	26	44	24	17	873	4	4	1
Horse	7	0	17	34	49	49	2	836	2	4
Ship	39	9	4	10	5	2	0	2	913	16
Truck	28	33	4	19	4	7	3	10	28	864

Predicted Class

Confusion Matrix



Actual Class

Predicted Class

	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
Plane	855	9	15	16	16	7	6	7	50	19
Car	17	886	3	6	2	1	5	3	32	45
Bird	52	2	737	46	61	46	32	13	11	0
Cat	16	1	46	713	45	111	35	14	13	6
Deer	6	0	31	41	853	18	20	27	3	1
Dog	7	1	20	124	28	783	10	21	3	3
Frog	6	1	26	44	24	17	873	4	4	1
Horse	7	0	17	34	49	49	2	836	2	4
Ship	39	9	4	10	5	2	0	2	913	16
Truck	28	33	4	19	4	7	3	10	28	864

Demo