

# DogVCat: Create an algorithm to distinguish dogs from cats

Presenter: Bang Liu, Yan Liu, Kai Zhou

Instructor: Dr Russ Greiner


Co-coach: Junfeng Wen

# Outlines

- ❏ Our Task
- ❏ Our Solutions
  - ❏ Traditional Model
  - ❏ Deep Learning Model
- ❏ Our Results
- ❏ Future Work

# Our Task

## ❖ Basic task

- To Create an algorithm to classify whether an image contains a dog or a cat
- Kaggle Competition 

## ❖ Dataset: the ASIRRA dataset provided by Microsoft Research

- Training Dataset: 543.76MB, 25000 images(12500 dogs, 12500 cats)
- Testing Dataset: 271.15MB, 12500 images

# Our Task

## ❖ Input & Output

- Input: images of **dogs** and **cats**
- Output: classification accuracy on testing dataset

## ❖ Relevance to Machine Learning

- Learning task
  - For **training dataset**, to **learn a classification model** to determine the decision boundary.
- Performance task
  - For **testing dataset**, to **make classification** for each image based on the learned model, and get the **accuracy**.

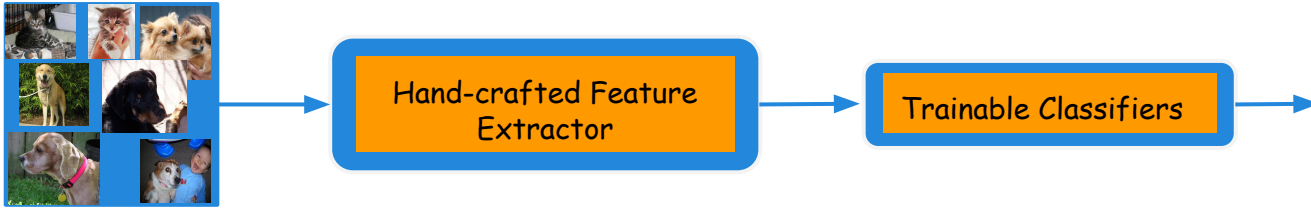
# Outlines

- ❏ Our Task
- ❏ **Our Solutions**
  - ❏ Traditional Model
  - ❏ Deep Learning Model
- ❏ Our Results
- ❏ Future Work

# Our Solutions

## ❖ Traditional Model for pattern recognition

### ➤ Fixed features + Trainable classifiers



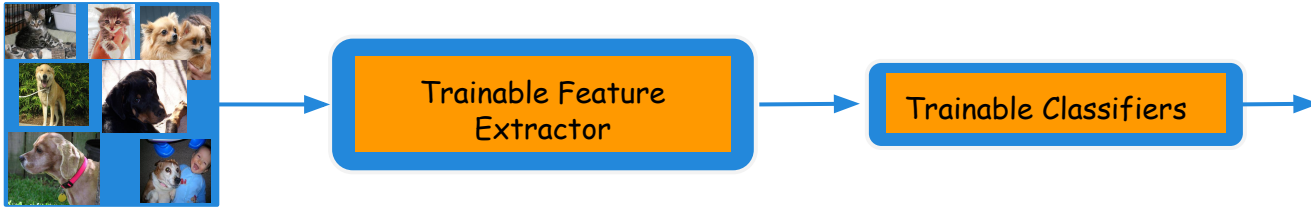
### ➤ Fixed features

- Low Level Features: **Color**
- High Level Features: **Dense SIFT** (Scale-Invariant Feature Transform)

# Our Solutions

## ❖ Deep Learning Model

- Trainable features + Trainable classifiers



- Trainable features
  - from Deep Neural Networks

# Our Solutions

## ❖ Two trainable classifiers

based on **data property** (high dimensionality) and **previous work**

- Support Vector Machines (SVMs)
- Deep Neural Networks



# Outlines

- ❏ Our Task
- ❏ Our Solutions
  - ❏ Traditional Model
  - ❏ Deep Learning Model
- ❏ Our Results
- ❏ Future Work

# Solution 1: Traditional Model

## ❖ Fixed Features

### ➤ High Level Features

- Dense SIFT (Scale-Invariant Feature Transform)

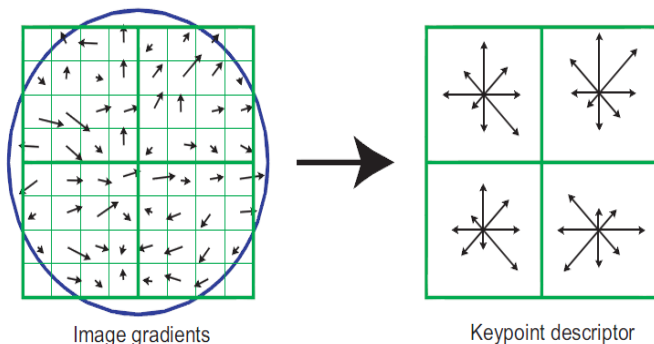
### ➤ Low Level Features

- Color

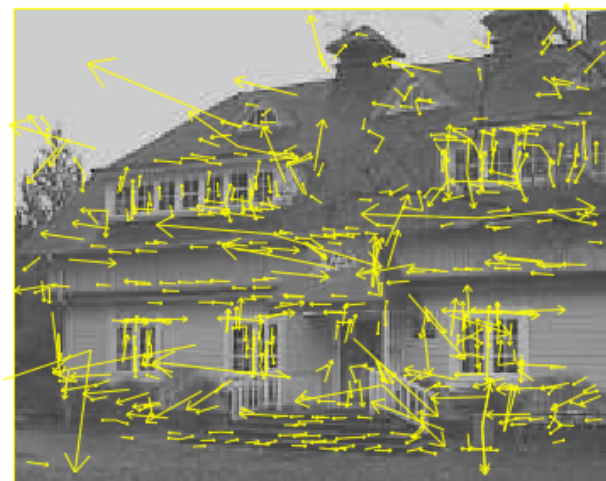
# Solution 1: Traditional Model

## ❖ Feature Extraction - (Dense) SIFT feature

- The SIFT features are local and based on the appearance of the object
- **Invariant** to image **scale** and **rotations**



61.6%



SIFT(ijcv04)

# Solution 1: Traditional Model

## ❖ Feature Extraction - Color

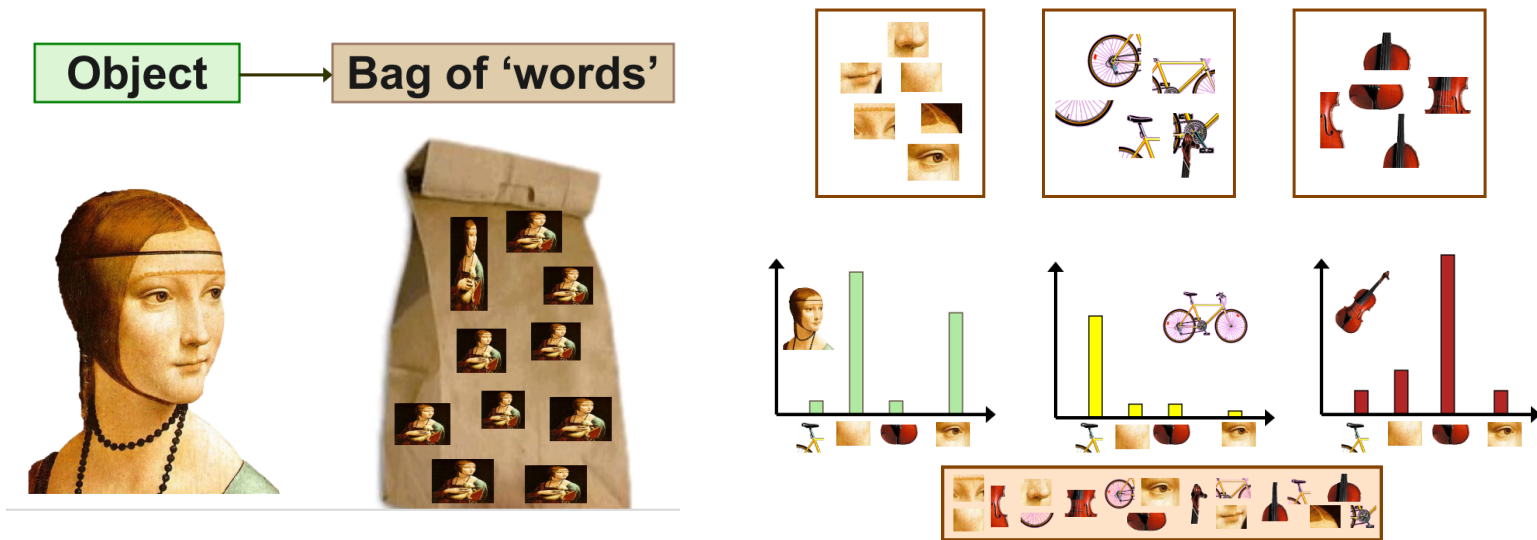


- Use **HSV** (hue, saturation, value) model other than **RGB** (red, green, blue) model of color
  - Closer to human perception of color
  - Easier to interpret

61.6% → 71.5%

# Solution 1: Traditional Model

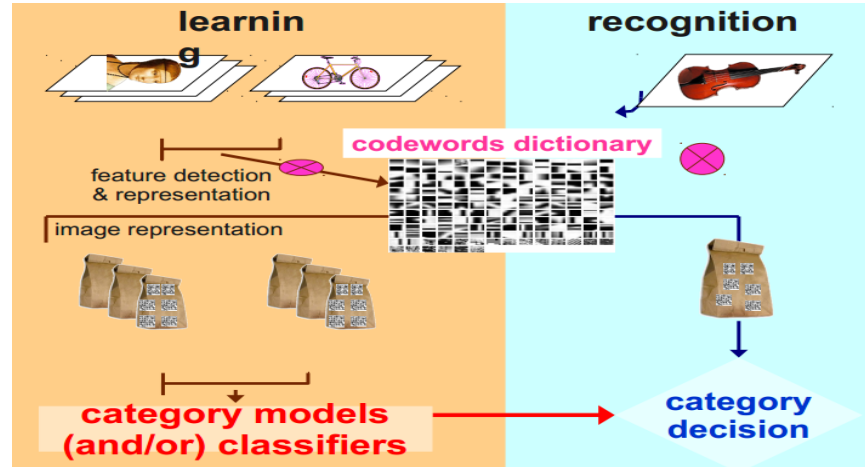
## ❖ Feature Representation - Bag of Words



# Solution 1: Traditional Model

## ❖ Feature Extraction - Whole Process

1. feature detection
2. create dictionary
3. representation
4. training model
5. get decision boundary



# Solution 1: Traditional Model

## ❖ Preprocessing: Image Segmentation

- Segmentation of pets from backgrounds  
-- complicated and various backgrounds

- Results (good & bad):



- Performance: **no improvement**

# Outlines

- ❏ Our Task
- ❏ **Our Solutions**
  - ❏ Traditional Model
  - ❏ **Deep Learning Model**
- ❏ Our Results
- ❏ Future Work



# Solution 2: Deep Learning

- ❖ Representations are hierarchical and trained



Low-Level  
Features

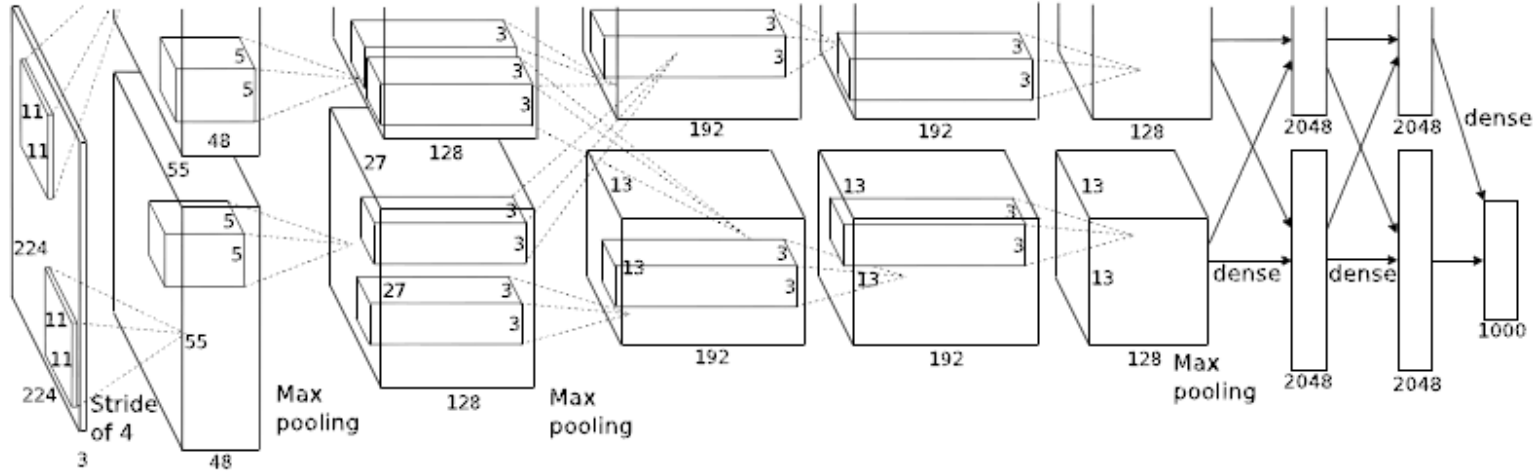
Mid-Level  
Features

High-Level  
Features

Trainable  
Classifier

# Solution 2: Deep Learning

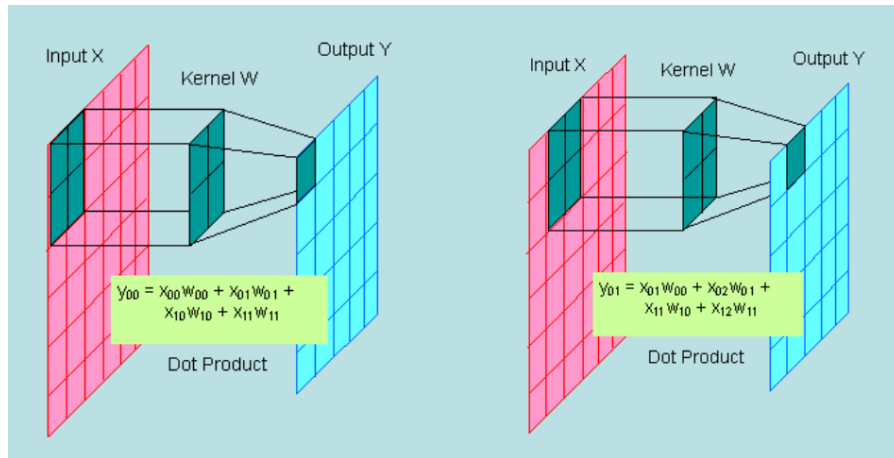
## ❖ Architecture of model: previous work



(Alex Krizhevsky, ImageNet 2012, 60 million parameters, 650,000 neurons, trained on 1.2 million training images)

# Solution 2: Deep Learning

## ❖ Convolutional layers: feature maps



1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

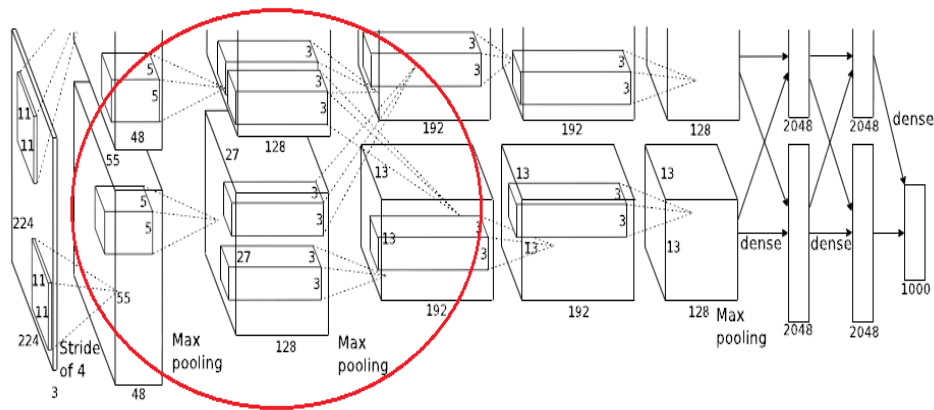
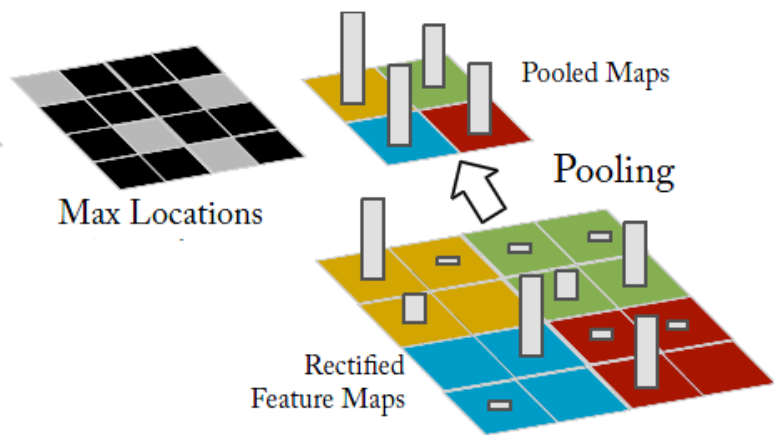
Image

4		

Convolved  
Feature

# Solution 2: Deep Learning

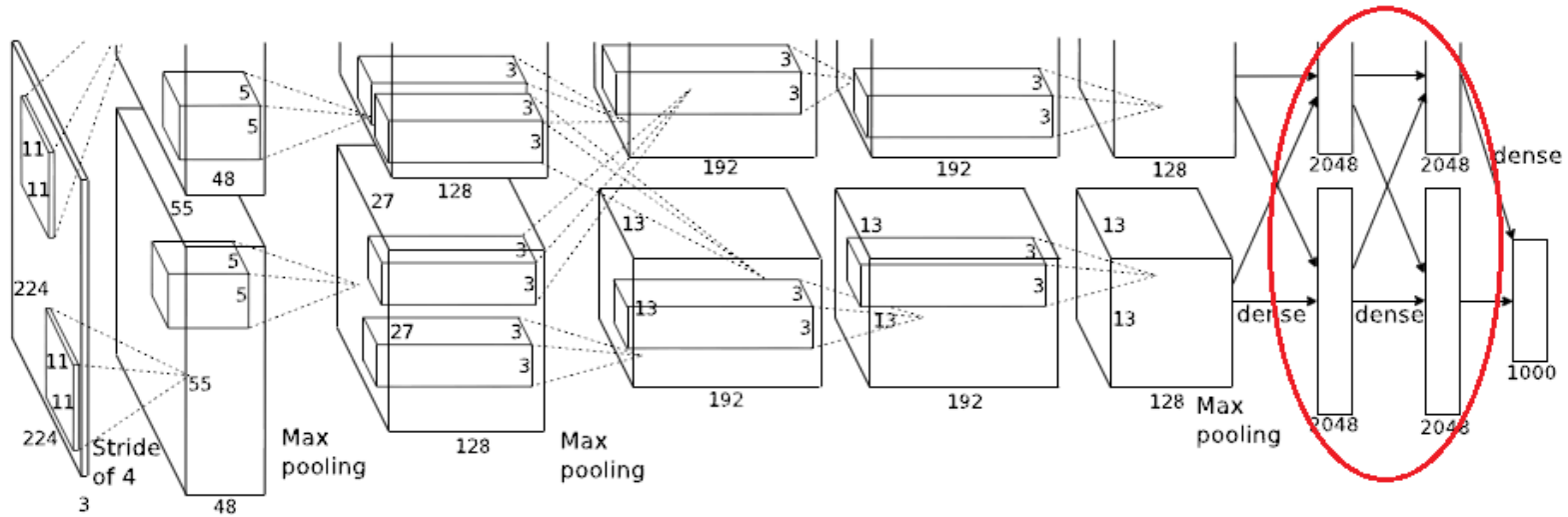
## ❖ Max pooling: sub-sampling



(Matthew D. Zeiler, 2013)

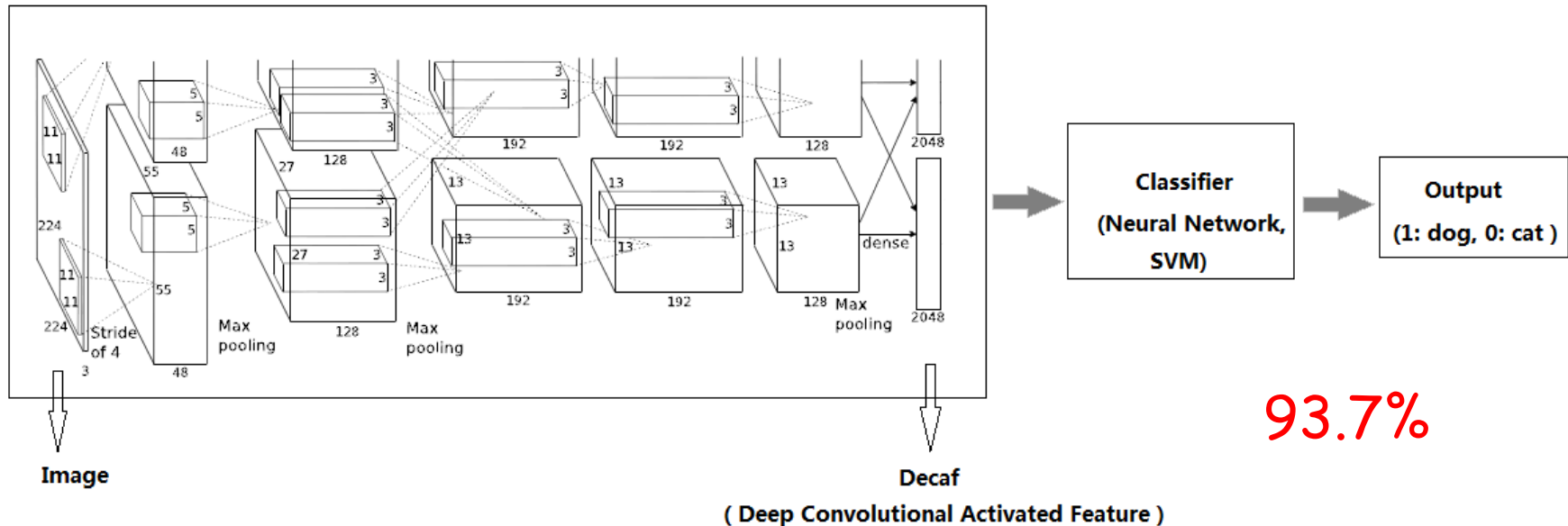
# Solution 2: Deep Learning

## ❖ Fully connected layers: image representation



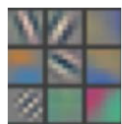
# Solution 2: Deep Learning

❖ Our model: Learned feature + Classifier

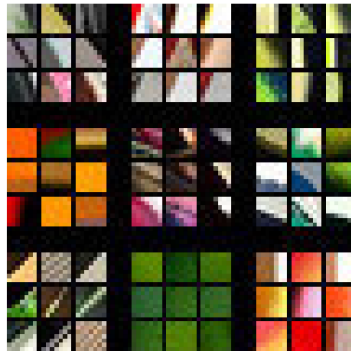


# Solution 2: Deep Learning

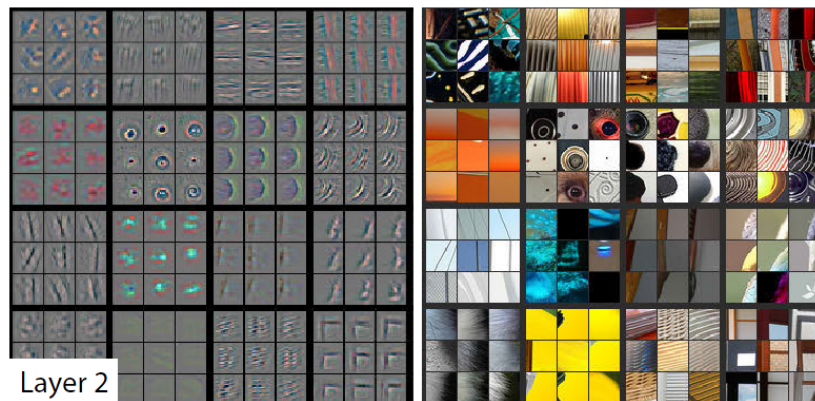
## ❖ Top activation for Layer 1, 2



Layer 1



edges, colors (low-level feature)



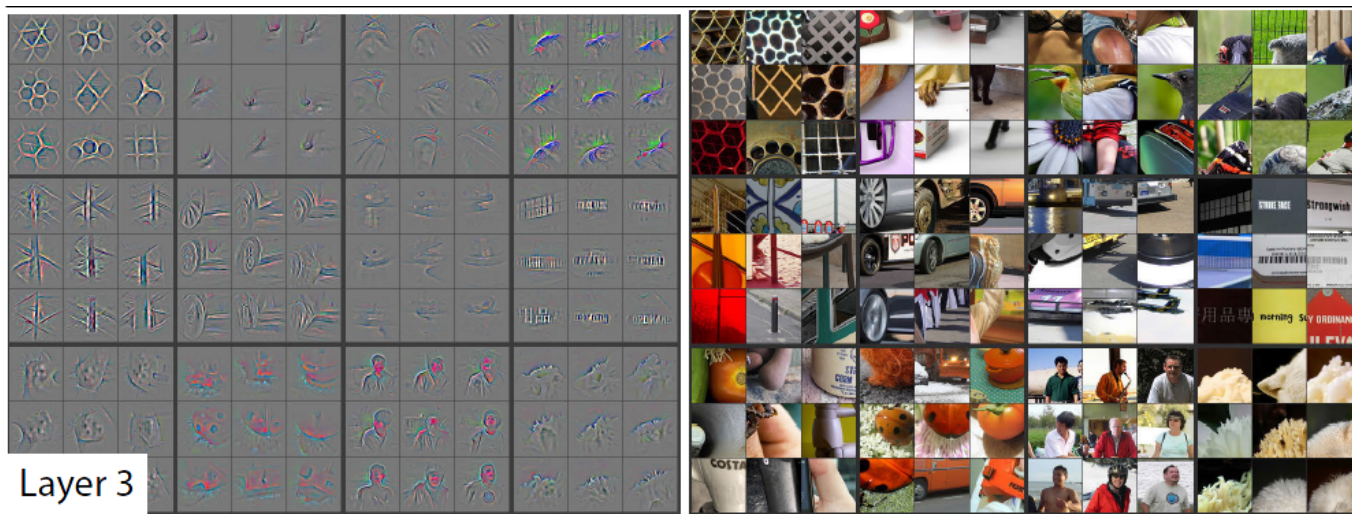
Layer 2

corners, edge/color conjunctions (low-level feature)

(Matthew D. Zeiler, 2013)

# Solution 2: Deep Learning

## ❖ Top activation for Layer 3



textures (mid-level feature)

(Matthew D. Zeiler, 2013)



# Solution 2: Deep Learning

## ❖ Top activation for Layer 4

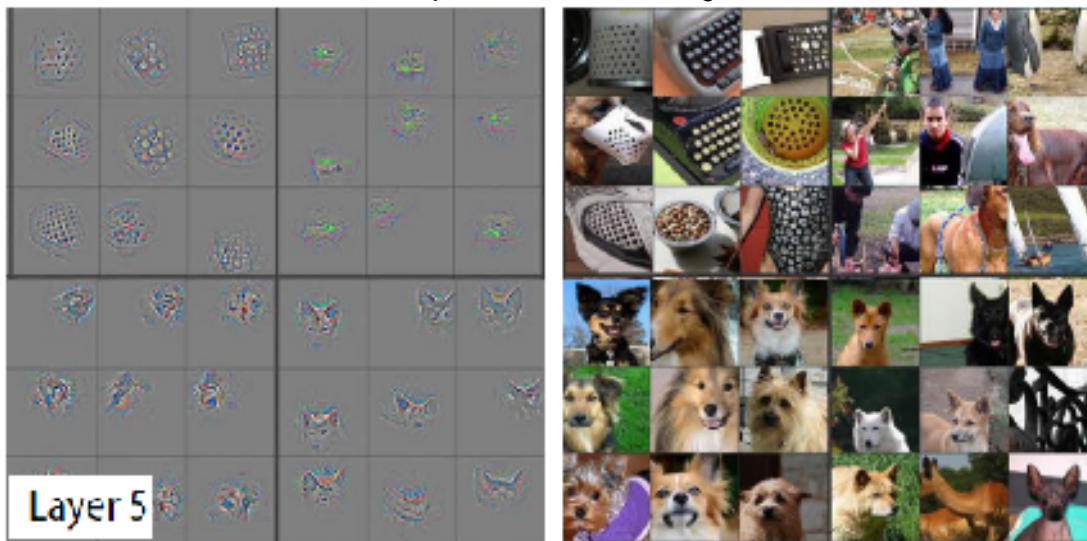


part of objects (high-level feature)

(Matthew D. Zeiler, 2013)

# Solution 2: Deep Learning

## ❖ Top activation for Layer 5 (objects)



entire objects (high-level feature)

(Matthew D. Zeiler, 2013)

# Outlines

- ❏ Our Task
- ❏ Our Solutions
  - ❏ Traditional Model
  - ❏ Deep Learning Model
- ❏ **Our Results**
- ❏ Future Work

# Our Results

## ❖ Performance on Fixed Features

### ➤ Fixed features + SVMs

- Dense SIFT features

0.61804 → 0.65067 → 0.67600

- Dense SIFT + Color features

0.54960 → 0.55867 → 0.71467

- Dense SIFT features(with segmentation)

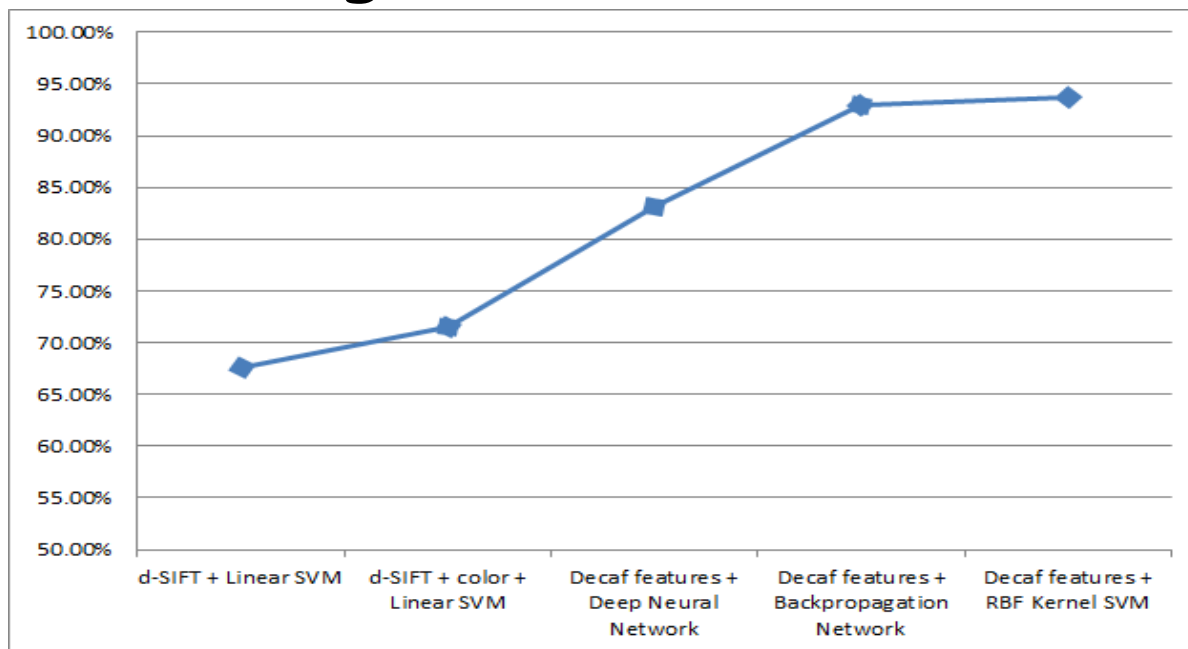
0.49600 → 0.50133

# Our Results

- ❖ **Performance** on Trainable Features
  - **Trainable Features + Deep Neural Network**
    - Original neural network(Alex 2012)  
0.83200
    - Decaf features + BP Neural Network  
0.93013
    - Decaf features + SVM (RBF Kernel)  
0.93787

# Our Results

## ❖ Increasing Performance



# Our Results

❖ Leaderboard on Kaggle (ranked **9th** in **71** teams)

#	Δ1w	Team Name <small>* in the money</small>	Score <small>🔒</small>	Entries	Last Submission UTC (Best - Last Submission)
1	-	DaggerFS *	<a href="#">0.97040</a>	4	<a href="#">Mon, 18 Nov 2013 17:54:09 (-3d)</a>
2	-	Charlie *	<a href="#">0.96987</a>	5	<a href="#">Sat, 16 Nov 2013 17:30:20 (-30.6d)</a>
3	-	Jeff	<a href="#">0.96773</a>	2	<a href="#">Thu, 26 Sep 2013 13:54:31</a>
4	-	wqren	<a href="#">0.96667</a>	4	<a href="#">Wed, 23 Oct 2013 02:12:19 (-1.1h)</a>
5	-	Kyle Kastner	<a href="#">0.96667</a>	10	<a href="#">Mon, 25 Nov 2013 15:17:11 (-4.1d)</a>
6	-	Daniel Nouri	<a href="#">0.96587</a>	8	<a href="#">Fri, 29 Nov 2013 19:11:26 (-10.9d)</a>
7	↑1	hungry red panda	<a href="#">0.96427</a>	2	<a href="#">Wed, 27 Nov 2013 09:17:39</a>
8	↓1	naxeji	<a href="#">0.94933</a>	10	<a href="#">Thu, 24 Oct 2013 19:00:57 (-21h)</a>
9	↑5	<b>UA_551</b> <small>🏆</small>	<a href="#">0.93787</a>	<b>18</b>	<a href="#">Tue, 03 Dec 2013 01:55:13 (-0.7h)</a>
10	↓1	Poly <small>🏆</small>	<a href="#">0.92000</a>	4	<a href="#">Sun, 17 Nov 2013 09:19:38 (-4d)</a>

# Outlines

- ❏ Our Task
- ❏ Our Solutions
  - ❏ Traditional Model
  - ❏ Deep Learning Model
- ❏ Our Results
- ❏ **Future Work**



# Future Work

## ❖ What we are going to do...

- to achieve **higher performance**
  - different **parameter settings** for SVMs and Deep Neural Networks
  - **Features combination**  
thanks to **Dr Russ Greiner**
  - **Face detection**
  - **multi classifiers** for different types of images  
thanks to **Dr Mohamed Elgendi**

# New Ideas

## ❖ Done

### ➤ Works:

- Combined **SIFT** with **Color** features
- Tried different types of **Decaf** features
- **SVMs** on Decaf features

### ➤ Doesn't work:

- Image Segmentation

## ❖ To Do

- Combine **Decaf** features with other features
- **Face Detection**

# References

- [1] Zeiler, Matthew D., and Rob Fergus. "Visualizing and Understanding Convolutional Neural Networks." *arXiv preprint arXiv:1311.2901* (2013).
- [2] Donahue, Jeff, et al. "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition." *arXiv preprint arXiv:1310.1531* (2013).
- [3] Krizhevsky, Alex, Ilya Sutskever, and Geoff Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in Neural Information Processing Systems* 25. 2012.
- [4] Lowe, David G. "Distinctive image features from scale-invariant keypoints." *International journal of computer vision* 60.2 (2004): 91-110.

# DogVCat

