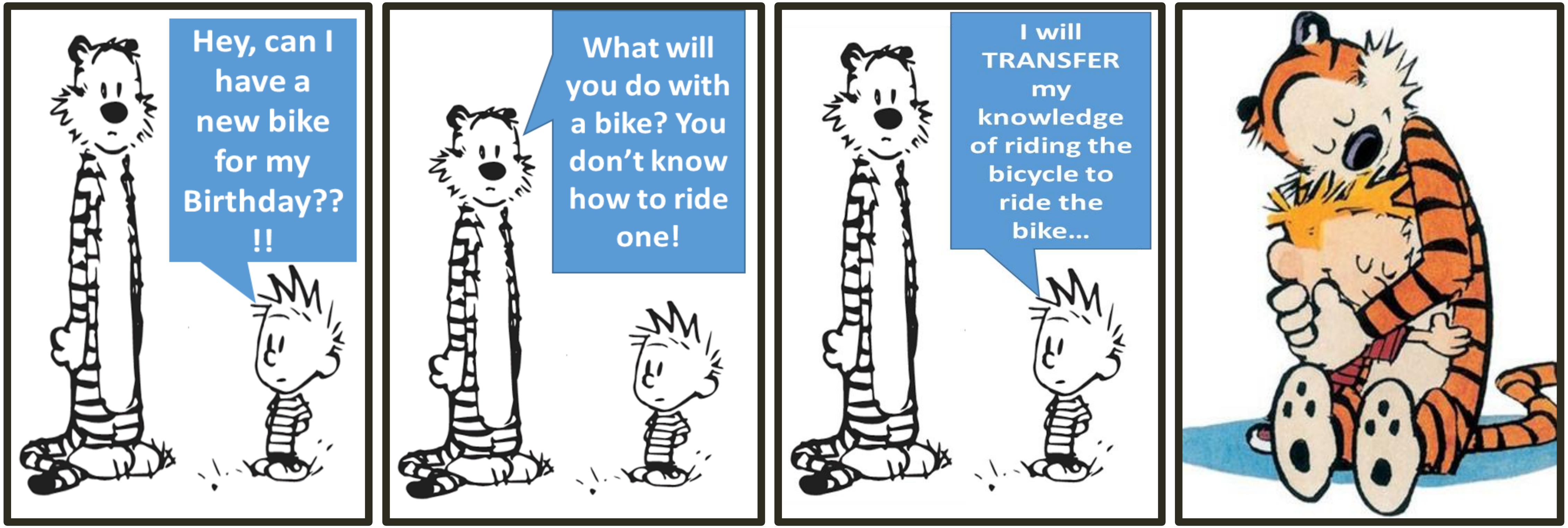
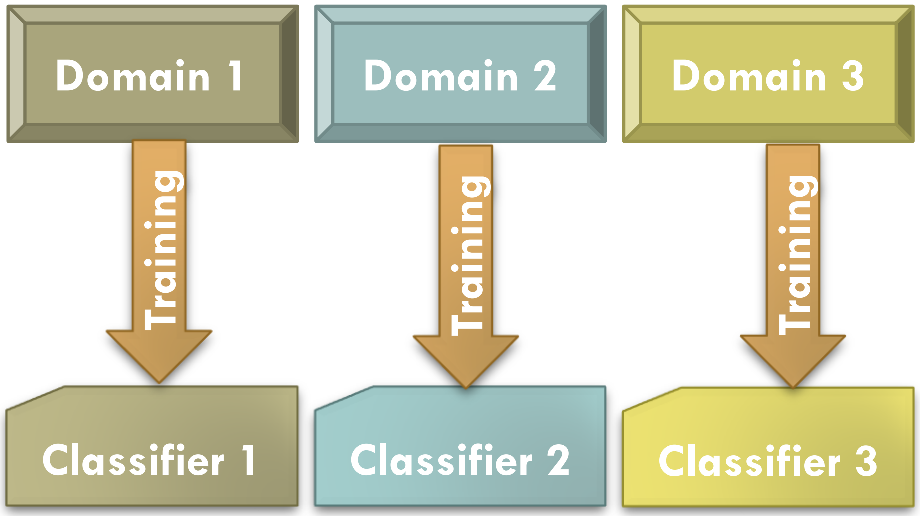
TRANSFER LEARNING FOR IMAGE CLASSIFICATION

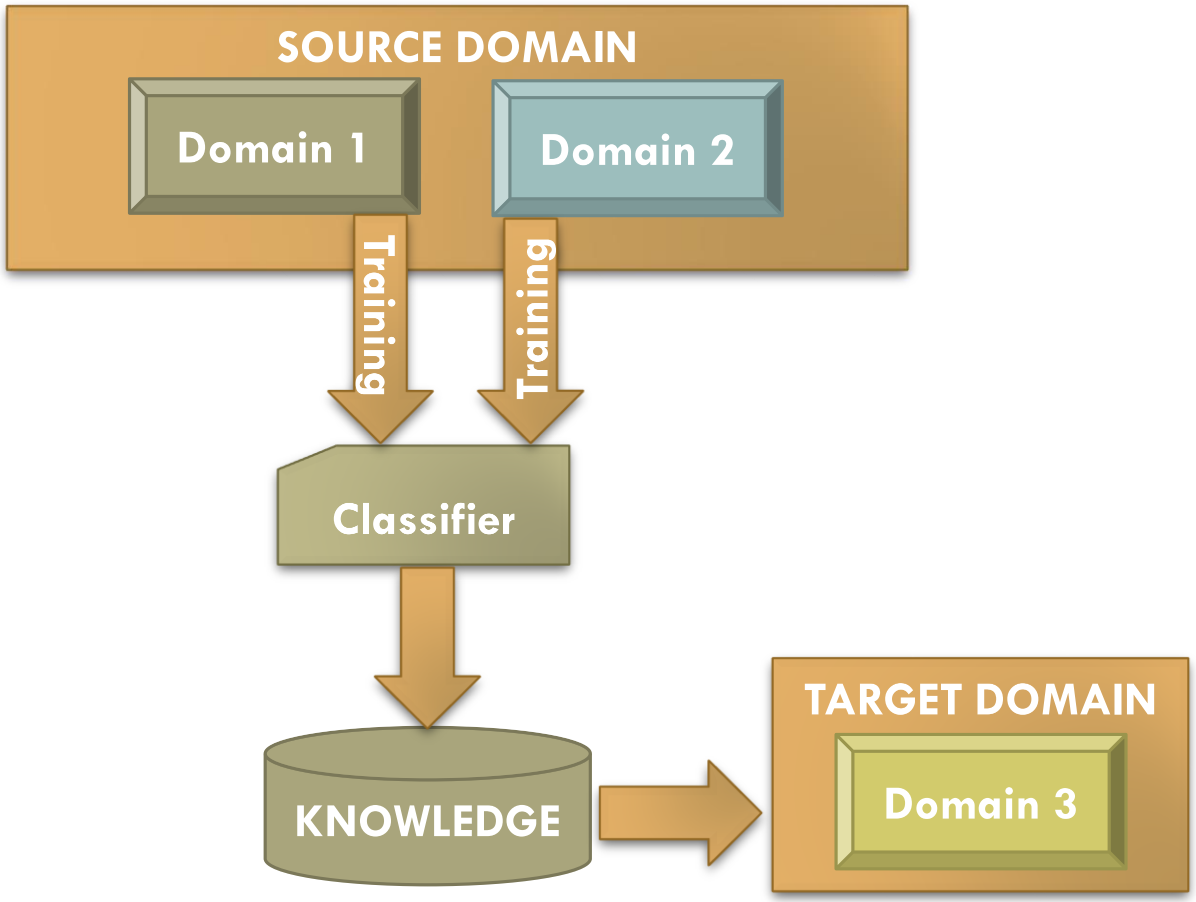
Arihant Jain
Siddharth Srivastava
Sumit Soman

WHAT IS TRANSFER LEARNING?



TRADITIONAL MACHINE LEARNING V/S TRANSFER LEARNING





PROBLEM UNDER STUDY

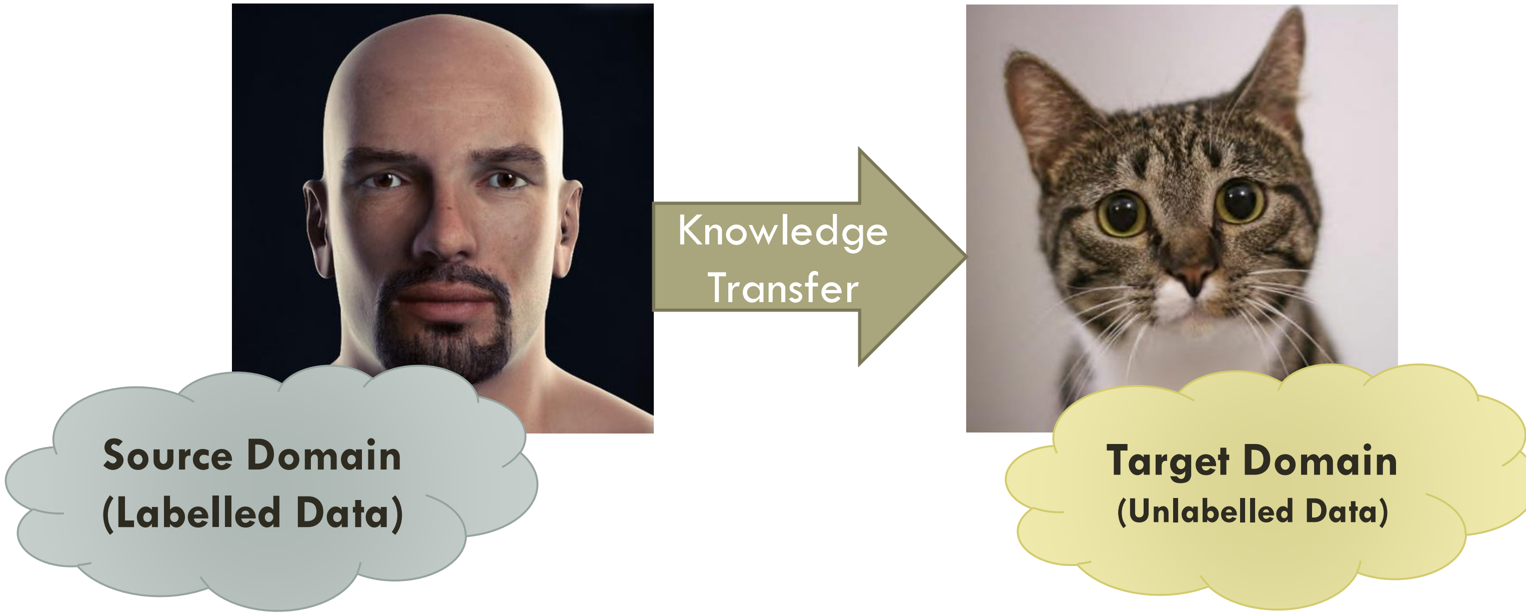
Attempt to classify images of human faces (source domain) V/s cat faces (target domain).

Initial experiments on Caltech 101 dataset

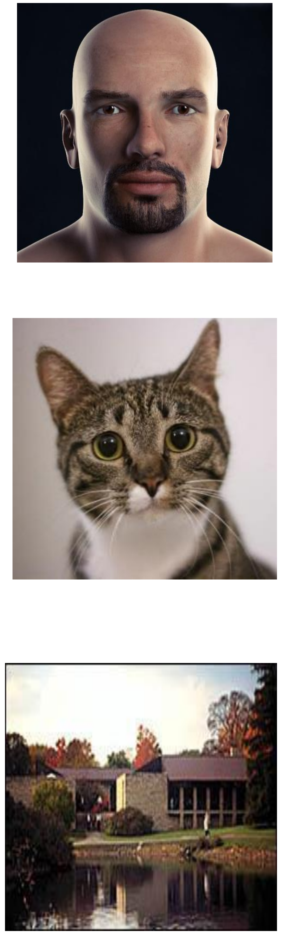
Dataset created from Google Images

- 100 images of human face, cat face and non-faces
- Images converted to grayscale
- Images resized to 128X128

Challenges in dataset: Images in different sizes, orientations, forms of object etc.



FEATURES: USING SURF

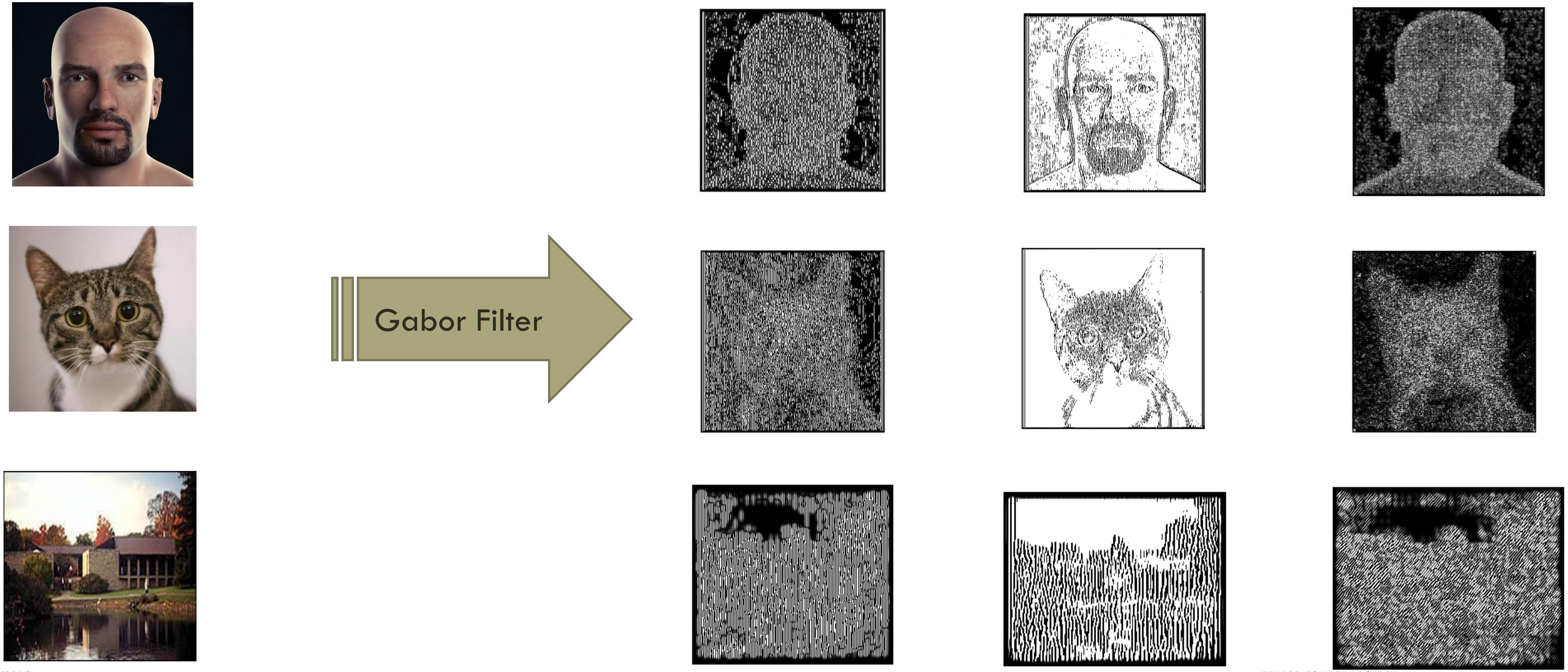


SURF Features

Strongest Features

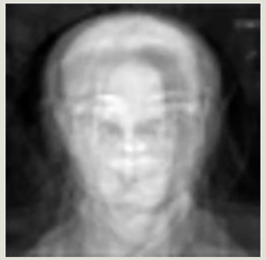
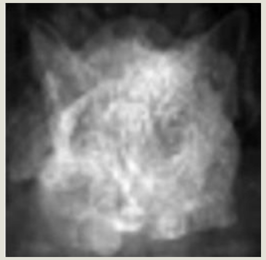
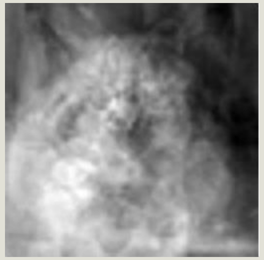
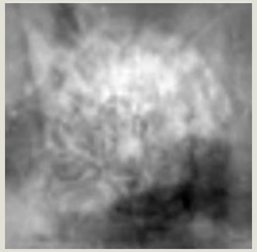
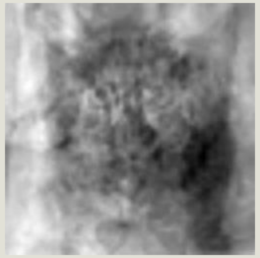
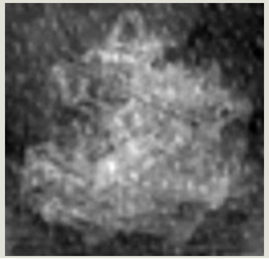
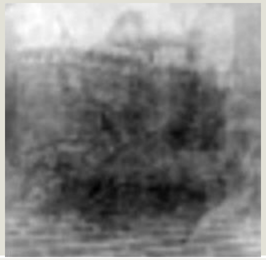
32 64 128

FEATURES: USING GABOR FILTER

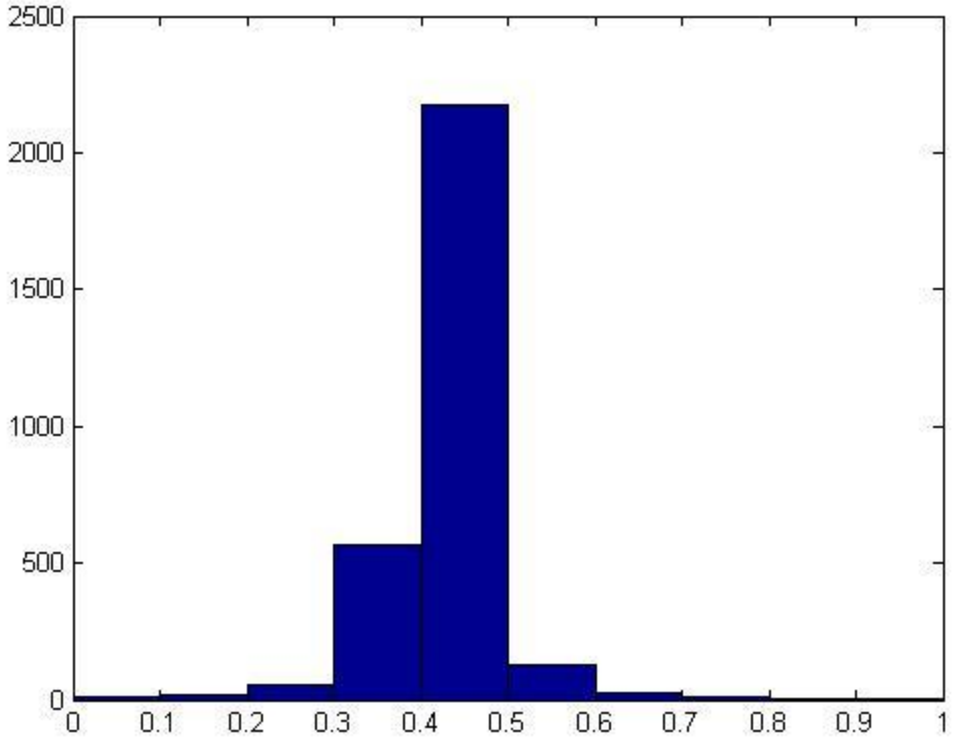


FEATURES: USING PCA

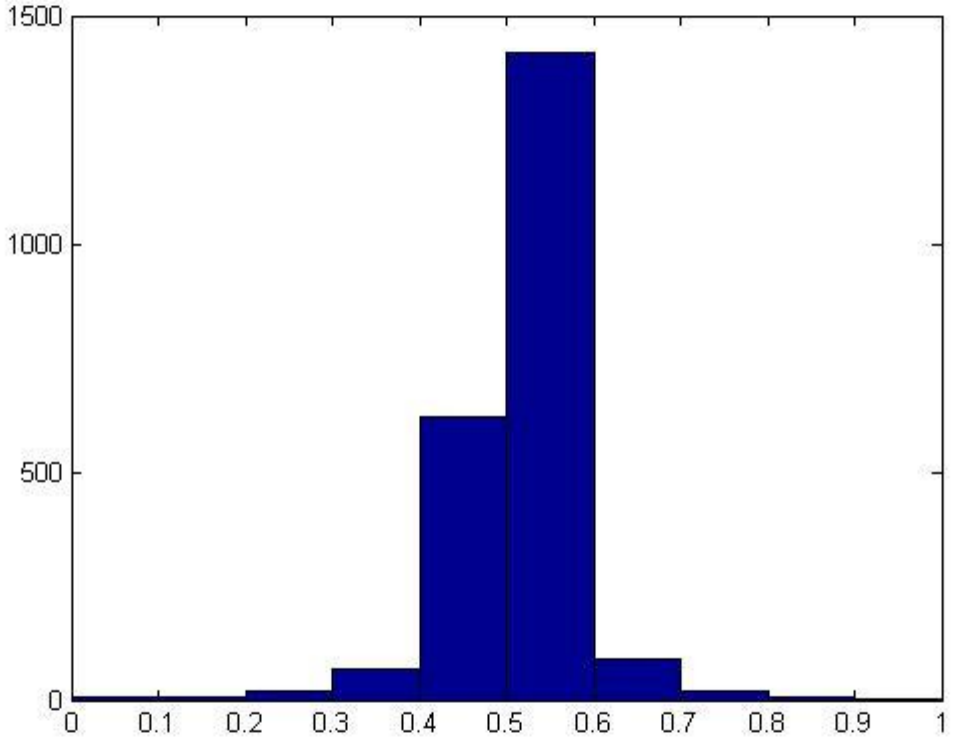
Principal Components Analysis

PC1	PC2	PC3	PC4
			
1.04 X 10^7**(51.12%)**	0.24 X 10^7**(12.13%)**	1.35 X 10^7**(6.70%)**	0.09 X 10^7**(4.47%)**
			
6.65 X 10^6**(33.93%)**	2.97 X 10^6**(15.13%)**	1.61 X 10^6**(8.23%)**	1.37 X 10^6**(7.07%)**
			
6.95 X 10^6 **(35.76%)**	2.83 X 10^6**(14.57%)**	1.69 X 10^6**(8.73%)**	1.39 X 10^6**(7.16%)**

DISTRIBUTION OF FEATURES



Face features



Cat features

TRANSFER LEARNING

ADAPTIVE SVM

Delta Function

$$f(\mathbf{x}) = f^a(\mathbf{x}) + \Delta f(\mathbf{x}) = f^a(\mathbf{x}) + \mathbf{w}^T\phi(\mathbf{x})$$

Formulation of A-SVM

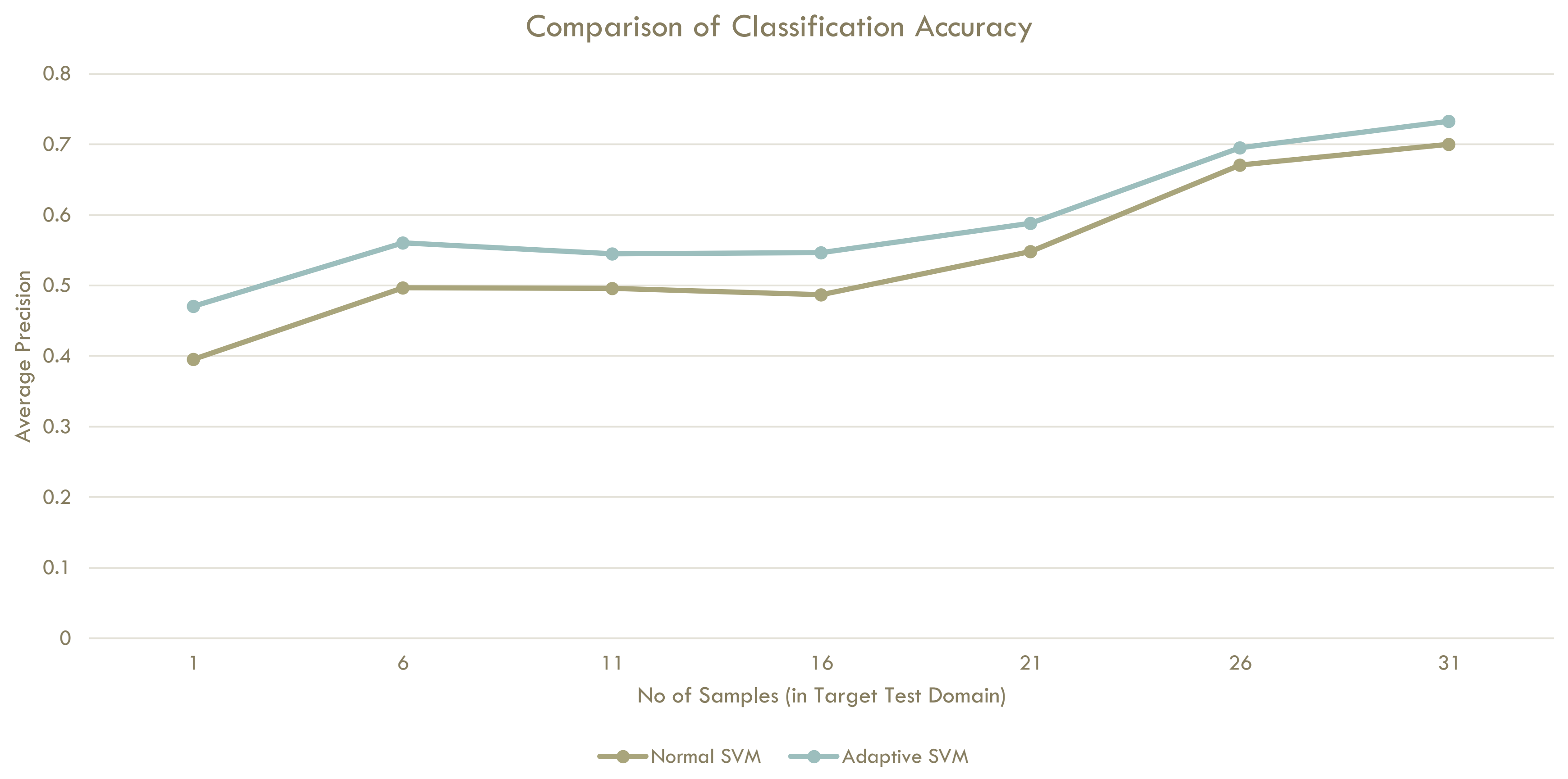
$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{s.t.} \quad \xi_i \geq 0$$

$$y_i f^a(\mathbf{x}_i) + y_i\mathbf{w}^T\phi(\mathbf{x}_i) \geq 1 - \xi_i, \quad \forall(\mathbf{x}_i, y_i) \in \mathcal{D}_l^p$$

Knowledge Transfer

RESULTS & COMPARISON WITH TRADITIONAL LEARNING



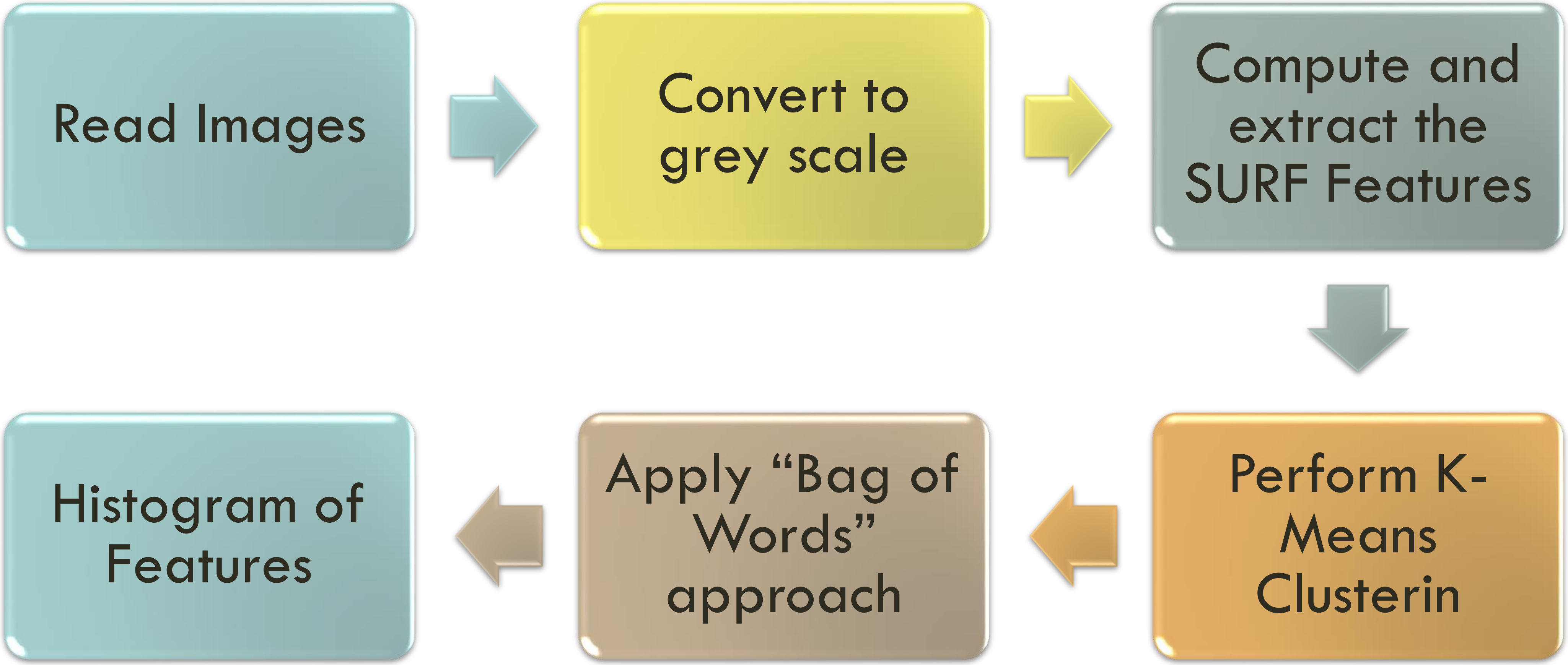
FUTURE WORK AND APPLICATIONS

Suggested applications of Transfer Learning include:

- Classification of new videos on YouTube, using knowledge transfer between domains.
- Classification of Sports/Non-sport Videos using texture-based features such as Gabor filter.
- Identifying new domains in YouTube videos, e.g. Harlem Shake

THANK YOU

FEATURE CALCULATION USING SURF



FEATURE CALCULATION USING GABOR FILTER

Set the Parameters like orientation , wavelength etc.

Read Images (Grey Scale)

Resize the Image to 128x128

Compute the Gabor filter window

Convolution

Calculate the Mean and Standard Deviation of Output Image

Complex

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = \exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)\exp\left(i\left(2\pi\frac{x'}{\lambda}+\psi\right)\right)$$

Real

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = \exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda}+\psi\right)$$

Imaginary

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = \exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)\sin\left(2\pi\frac{x'}{\lambda}+\psi\right)$$

where

$$x' = x\cos\theta + y\sin\theta$$

and

$$y' = -x\sin\theta + y\cos\theta$$

FEATURE EXTRACTION USING PCA

Get data as [features(M) X samples(N)]

$$u[m] = \frac{1}{N}\sum_{n=1}^{N} X[m,n]$$

Find the empirical mean od he data.

$$\mathbf{B} = \mathbf{X} - \mathbf{u}\mathbf{h}$$

$$h[n] = 1 \qquad \text{for } n = 1, \ldots, N$$

Subtract the mean from the data

$$\mathbf{C} = \frac{1}{N-1}\mathbf{B} \cdot \mathbf{B}^{*}$$

Compute the covariance matrix

Calculate the eigenvectors and eigenvalues of the covariance matrix

$$\mathbf{V}^{-1}\mathbf{C}\mathbf{V} = \mathbf{D}$$

Select a subset of the eigenvectors as basis vectors

W[p,q]=V[p,q] for p=1….M, q=1….L

Project data into k-space (lower dimensional space)

Y=W*X

NOTATION

Domain:

It consists of two components: A feature space $\mathcal{X}$, a marginal distribution

$\mathcal{P}(X)$, where $X = \{x_1, x_2, ..., x_n\} \in \mathcal{X}$

In general, if two domains are different, then they may have different feature spaces or different marginal distributions.

Task:

Given a specific domain and label space $\mathcal{Y}$, for each x_i in the domain, to predict its corresponding label y_i, where $y_i \in \mathcal{Y}$

In general, if two tasks are different, then they may have different label spaces or different conditional distributions

$\mathcal{P}(Y|X)$, where $Y = \{y_1, ..., y_n\}$ and $y_i \in \mathcal{Y}$

NOTATION

For simplicity, we only consider at most two domains and two tasks.

Source domain:

$$\mathcal{P}(X_S), \text{ where } X_S = \{x_{S_1}, x_{S_2}, ..., x_{S_{n_S}}\} \in \mathcal{X}_S$$

Task in the source domain:

$$\mathcal{P}(Y_S|X_S), \text{ where } Y_S = \{y_{S_1}, y_{S_2}, ..., y_{S_{n_S}}\} \text{ and } y_{S_i} \in \mathcal{Y}_S$$

Target domain:

$$\mathcal{P}(X_T), \text{ where } X_T = \{x_{T_1}, x_{T_2}, ..., x_{T_{n_T}}\} \in \mathcal{X}_T$$

Task in the target domain

$$\mathcal{P}(Y_T|X_T), \text{ where } Y_T = \{y_{T_1}, y_{T_2}, ..., y_{T_{n_T}}\} \text{ and } y_{T_i} \in \mathcal{Y}_T$$