

**ANALYZING THE EFFECTS OF LOW-LEVEL FEATURES  
FOR VISUAL ATTRIBUTE RECOGNITION**

**GÖRSEL NİTELİK ÖĞRENMEDE ALT-DÜZEY  
ÖZNİTELİKLERİN ETKİLERİNİN ANALİZİ**

**EMİNE GÜL DANACI**

**ASST. PROF. DR. NAZLI İKİZLER-CİNBİŞ**

**Supervisor**

Submitted to Graduate School of Science and Engineering of Hacettepe University  
as a Partial Fulfillment to the Requirements  
for the Award of the Degree of Master of Science  
in Computer Engineering

September 2015

This work named "ANALYZING THE EFFECTS OF LOW-LEVEL FEATURES FOR VISUAL ATTRIBUTE RECOGNITION" by EMİNE GÜL DANACI has been approved as a thesis for the Degree of MASTER OF SCIENCE IN COMPUTER ENGINEERING by the below mentioned Examining Committee Members.

Assoc. Prof. Dr. Pınar DUYGULU ŞAHİN  
Head



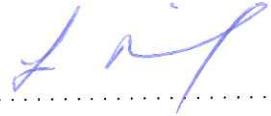
Asst. Prof. Dr. Nazlı İKİZLER CİNBİŞ  
Supervisor



Asst. Prof. Dr. M. Erkut ERDEM  
Member



Asst. Prof. Dr. Selen PEHLİVAN  
Member



Asst. Prof. Dr. Ufuk ÇELİKCAN  
Member



This thesis has been approved as a thesis for the Degree of MASTER OF SCIENCE IN COMPUTER ENGINEERING by Board of Directors of the Institute for Graduate School of Science and Engineering.

Prof. Dr. Fatma SEVİN DÜZ  
Director of the Institute of  
Graduate School of Science and Engineering

## ETHICS

In this thesis study, prepared in accordance with the spelling rules of Graduate School of Science and Engineering of Hacettepe University,

I declare that

- all the information and documents have been obtained in the base of the academic rules.
- all audio-visual and written information and results have been presented according to the rules of scientific ethics
- in case of using others works, related studies have been cited in accordance with the scientific standards
- all cited studies have been fully referenced
- I did not do any distortion in the data set
- and any part of this thesis has not been presented as another thesis study at this or any other university.

20/08/2015



EMİNE GÜL DANACI

## **ABSTRACT**

### **ANALYZING THE EFFECTS OF LOW-LEVEL FEATURES FOR VISUAL ATTRIBUTE RECOGNITION**

**Emine Gül DANACI**

**Master of Science, Computer Engineering Department**

**Supervisor: Asst. Prof. Dr. Nazlı İKİZLER CİNBIŞ**

**September 2015, 80 pages**

In recent years, visual attributes became a popular topic of computer vision research. Visual attributes are being used on various tasks including object recognition, people search, scene recognition, and so on. In order to encode the visual attributes, a common applied procedure for supervised learning of attributes is to extract low-level visual features from the images first. Then, an attribute learning algorithm is applied and visual attribute models are formed.

In this thesis, we explore the effects of using different low-level features on learning visual attributes. For this purpose, we use various low-level features, which aim to capture different visual characteristics, such as shape, color and texture. In addition, we also evaluate the effect of the recently evolving deep features on the attribute learning problem. Experiments have been carried out on four different datasets, which were collected for different visual recognition tasks and extensive evaluations have been reported. Our results show that, while using the supervised deep features are effective, using them in combination with low-level features are more effective for visual attribute learning.

**Keywords:** Visual attributes, low-level features, Texton, LBP, HOG, SIFT, CSIFT, CNN

## ÖZET

# GÖRSEL NİTELİK ÖĞRENMEDE ALT-DÜZEY ÖZİNİTELİKLERİN ETKİLERİNİN ANALİZİ

**Emine Gül DANACI**

**Yüksek Lisans, Bilgisayar Mühendisliği**

**Danışman: Yrd. Doç. Dr. Nazlı İKİZLER CİNBİŞ**

**Ağustos 2015, 80 sayfa**

Görsel nitelikler bilgisayarlı görü alanında son zamanlarda popüler olmaya başlamış bir konudur. Görsel nitelikler nesne tanıma, insan arama, sahne tanıma gibi bir çok alanda kullanılmaktadır. Görsel niteliklerin öğreticiyle öğrenilebilmesi için ilk adım düşük seviyeli öz niteliklerin çıkartılmasıdır. Sonrasında görsel nitelik öğrenme algoritmaları uygulanarak görsel nitelik modelleri oluşturulur.

Bu tez çalışmasında düşük seviyeli öz niteliklerin görsel nitelik öğrenmeye etkisi araştırılmıştır. Bu amaçla şekil, renk ve doku gibi farklı görsel karakteristikleri tanımlayabilen çeşitli öz nitelikler kullanılmıştır. Ayrıca gitgide gelişmekte olan derin öz niteliklerin görsel nitelik öğrenmeye etkileri de değerlendirilmiştir. Deneyleri gerçekleştirmek için farklı görsel tanıma görevleri için tanımlanmış dört adet veri kümesi kullanılmış ve sonuçları kaydedilmiştir. Sonuçlarımıza göre görsel nitelik öğrenme için derin öz nitelik kullanımı etkilidir. Bunun yanında bu öz niteliklerin düşük seviyeli öz nitelikler ile kombinasyonu daha etkili sonuçlar vermiştir.

**Anahtar Kelimeler:** Görsel nitelikler, alt-düze öz nitelikler, Texton, LBP, HOG, SIFT, CSIFT, CNN

## ***ACKNOWLEDGEMENTS***

This work was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) Career Development Award 112E149.

First of all, I would like to thank to my supervisor Asst. Prof. Dr. Nazlı İKİZLER CİNBIŞ for her valuable advice and guidance.

Besides I would like to thank to my thesis committee members, Doç. Dr. Pınar DUYGULU ŞAHİN, Yrd. Doç. Dr. Nazlı İKİZLER CİNBIŞ, Yrd. Doç. Dr. M. Erkut ERDEM, Yrd. Doç. Dr. Selen PEHLİVAN, Yrd. Doç. Dr. Ufuk ÇELİKCAN for reviewing this thesis and their helpful comments.

I appreciate for the support and encouragement of my family since the very beginning.

I would like to thank to my dear friend Rana ÖZAKINCI for her help and support. I would also like to thank to my all friends and colleagues for providing motivation and their good wishes.

I would also thank to my dear friend Aysun KOÇAK and members of Hacettepe University Computer Vision Laboratory(HUCVL) for their help and support.

I would also thank to my dear cousin Elif Sena AKKUŞ for being a companion during this thesis.

# CONTENTS

	<u>Page</u>
ABSTRACT .....	i
ÖZET .....	iii
ACKNOWLEDGEMENTS .....	iv
CONTENTS .....	v
FIGURES .....	vii
1. INTRODUCTION.....	1
1.1. Motivation .....	2
1.2. Major Contributions of This Thesis .....	3
1.3. Organization of the Thesis .....	3
2. BACKGROUND AND RELATED WORK .....	4
2.1. Attributes .....	4
2.2. Attribute Learning Methods .....	5
2.3. Tasks and Applications .....	11
2.4. Low-level Features Used in Attribute Learning .....	11
3. APPROACH .....	14
3.1. Feature Types.....	14
3.2. Method.....	19
3.3. Implementation Details .....	23
4. EXPERIMENTS & RESULTS .....	25
4.1. Datasets .....	25
4.2. Experiment Implementation Details .....	28
4.3. Experimental Results .....	28
4.4. Comparison with Reference Work .....	58
5. CONCLUSIONS .....	60
REFERENCES .....	61
CURRICULUM VITAE .....	70

## FIGURES

	<u>Page</u>
1.1. Examples of Attributes and Categories .....	2
3.1. Layers with kernel size of VGGNet Architecture .....	18
3.2. Approach of this work .....	20
4.1. Example images and their corresponding attributes for a-Pascal Dataset [1] ...	26
4.2. Example images and their corresponding attributes for a-Yahoo Dataset [1] ...	26
4.3. Example images and their corresponding attributes for Shoes Dataset [2].....	27
4.4. Example images and their corresponding attributes for Attributes of People Dataset [3] .....	28
4.5. ROC curves of low-level and mid-level features for a-Pascal dataset .....	33
4.6. ROC curves of low-level and mid-level features for a-Yahoo dataset .....	34
4.7. ROC curves of low-level and mid-level features for Shoes dataset .....	35
4.8. ROC curves of low-level and mid-level features for Attributes of People dataset	36
4.9. The results of attributes and low-level features and mid-level features corre- lations for a-Pascal dataset, Part-1.....	37
4.10. The results of attributes and low-level features and mid-level features corre- lations for a-Pascal dataset, Part-2.....	38
4.11. The results of attributes and low-level features and mid-level features corre- lations for a-Pascal dataset, Part-3.....	38
4.12. The results of attributes and low-level features and mid-level features corre- lations for a-Pascal dataset, Part-4.....	39
4.13. The results of attributes and low-level features and mid-level features corre- lations for a-Pascal dataset, Part-5.....	39
4.14. The results of attributes and low-level features and mid-level features corre- lations for a-Pascal dataset, Part-6.....	40



4.15. The results of attributes and low-level features and mid-level features correlations for a-Pascal dataset, Part-7.....	40
4.16. The results of attributes and low-level features and mid-level features correlations for a-Yahoo dataset, Part-1 .....	41
4.17. The results of attributes and low-level features and mid-level features correlations for a-Yahoo dataset, Part-2 .....	41
4.18. The results of attributes and low-level features and mid-level features correlations for a-Yahoo dataset, Part-3 .....	42
4.19. The results of attributes and low-level features and mid-level features correlations for a-Yahoo dataset, Part-4 .....	42
4.20. The results of attributes and low-level features and mid-level features correlations for a-Yahoo dataset, Part-5 .....	43
4.21. The results of attributes and low-level features and mid-level features correlations for Shoes dataset .....	44
4.22. The results of attributes and low-level features and mid-level features correlations for Attributes of People dataset.....	45
4.23. ROC curves of feature combinations for a-Pascal dataset - Part 1 .....	50
4.24. ROC curves of feature combinations for a-Pascal dataset - Part 2.....	51
4.25. ROC curves of late fusion feature combinations for a-Pascal dataset - Part1 ...	52
4.26. ROC curves of late fusion feature combinations for a-Pascal dataset - Part2 ...	53
4.27. ROC curves of wighted late fusion feature combinations for a-Pascal dataset..	54
4.28. ROC curves of feature combinations for a-Yahoo dataset .....	55
4.29. ROC curves of feature combinations for Shoes dataset .....	56
4.30. ROC curves of feature combinations for People dataset .....	57

## 1. INTRODUCTION

Computer vision algorithms are used for processing binary storage information of images and videos with the purpose of describing them in a human understandable way. In most of the computer vision applications, the first step of the process is to represent the images via low-level features like color histograms, histogram of orientations, edges, and more. Low-level features are used to describe or recognize objects directly. However, these features are inadequate for humans to understand since they only include machine detectable data.

Suppose that one intends to describe a bird. Low-level features can only detect basic information like its colors and size, but these features are not enough for describing particular properties such as feather, beak and head. These properties are called as *visual attributes* which give more information than low-level features. So, we can say the visual attributes are more meaningful than low-level features, and low-level features are used for learning these visual attributes.

Consider it from a different perspective; the image can be describable by only its category. For the example mentioned above, saying it is a bird can be enough in some cases. The label bird corresponds to the category label, as in the case of people, building, potted plant, and so on. Attributes are more detailed than these categories. So the attributes form a middle level between the low-level features and categories. Attributes can be any adjective, material and functional properties of objects. Glasses, head, face, hair, tail, plastic, black, white, stripes, natural, smiling, etc. can be example of attributes. Figure 1.1. shows example images, together with their category labels and list of attributes available in the images.

We need the attributes because the human vision sees the world with attributes. You can imagine a robot which wants to learn the way is walkable for him. If it can learn the attributes like muddy, then it can decide the road is walkable or not. It is inadequate only learnt the road. The attributes of road are also important. Because of these reasons, visual attribute learning has been included in the computer vision literature recently.

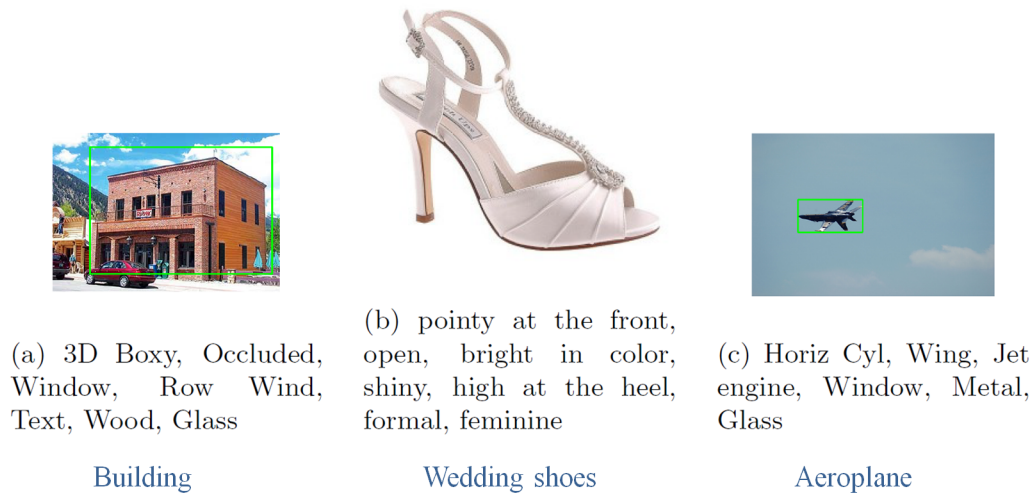


FIGURE 1.1.: Examples of Attributes and Categories

## 1.1. Motivation

Recently, visual attribute learning and usage have become a popular research topic of computer vision. Attributes have many application areas, including object recognition, face verification, product search, people search etc. Learning attributes automatically and using their expressive power is beneficial in the advancement of these applications.

In this thesis, we aim to explore which low-level features contribute to the modeling of the visual attributes the most. In this context, several low-level features that encode the color, texture and shape information in various levels are explored and their contribution to the recognition of the attributes are evaluated experimentally. To this end, we first evaluate the individual effects of low-level features in attribute learning, and then investigate the effect of using low-level features in combination. We experiment over four different datasets, including object recognition, shoes description, people description datasets. We choose these datasets in order to evaluate the effect of low-level features on attribute learning from different aspects.

In addition to evaluating the effect of using different low-level features, we also looked at the performance of recently evolving deep features, which can be considered as mid-level features which are shown to be quite effective in various computer vision tasks. In our experiments, we also evaluate their performance in attribute recognition scenarios and evaluate how they effect the performance when used alone or in combination with low-level features.

## **1.2. Major Contributions of This Thesis**

In this thesis, we have the following contributions :

- We evaluate the individual performances of various low-level features, including color, texture, shape in the attribute learning framework
- We evaluate the recent deep learning features [4] regarding their effect on attribute learning performance
- We evaluate the performance of the combination of these features on different datasets and different settings. We report individual findings on different attributes on different datasets.

To the best of our knowledge, there is no prior work that investigates the effect of low-level features on attribute recognition. So this work is intended to fill the deficiency of such comparisons in this field.

## **1.3. Organization of the Thesis**

In Chapter 2. we review the attribute learning methods. We give examples to these methods by explaining their classification methods and low-level features used in their works. In Chapter 3., we introduce the low-level features that we used in this work and explain the method with the usage of low-level features and classification method. In Chapter 4., we present results of low-level features and combined features. We explain the attribute and low-level features correlation and we compare our results with the-state-of-the-art methods. Lastly, in Chapter 5. we sum up the results and we discuss possible feature works.

## 2. BACKGROUND AND RELATED WORK

In this chapter, we first give a brief background about attributes and attribute recognition. Then, we present the related work on attribute learning methods, features used in attribute learning and applications.

### 2.1. Attributes

Attributes are depicted aspects of visual appearance which are human understandable as well as machine detectable. They have higher content than low-level features such as 'color', 'edge', and so on, but they have lower content than categories like 'cars', 'buildings', 'birds', and so forth. Attributes can be assumed within the range of low-level features and high-level categories such as 'natural', '3D', 'long hair', 'wooden' . Generally attributes can be expressed as any semantic, material or functional properties.

In current literature of computer vision, there is mainly three types of attributes that can be used for various applications which can be listed as binary, relative and spoken. We briefly define and review each of these attributes here.

**Binary Attributes** Many of the earliest works about visual attributes use binary attributes as their basis of representation [1], [5], [6]. Images comprise one or more objects where binary attributes are used to characterize these objects in a single image. An attribute can belong to an image or not, there is no other probability. For example, the person has some properties such as 'has sunglasses', 'has long hair', 'has t-shirt' which are called as attributes that is specific to this image. If we would try to discuss the emotions of the person in the picture, we can find whether the person is happy or sad, but we might not find strength information that indicates his emotions.

**Relative Attributes** The relative attribute concept was introduced to the field by Parikh and Grauman [7]. In their work, they stated that binary attributes can be inadequate for representing semantic relationships. Binary attributes cannot introduce comparisons between images. If we consider the example in the previous part we cannot compare the images of two people with binary attributes by using strength of their emotions. For such cases relative

## Chapter 1. *BACKGROUND AND RELATED WORK*

attributes are used to overcome this drawback. Based on this explanation, relative attributes can be defined as attributes such as 'more angry', 'more happy', 'more open', 'more broad', and so on. In this type of attributes, all images in the dataset are ranked by attributes. These attributes are useful when the purpose is comparing attribute magnitude with the choices 'more', 'less' or 'equal' and suitable for applications such as image search, people search, product search, and so on. If we want to find queries such as 'more angry person', 'less blond girl' these attributes are useful.

**Spoken Attributes** Sadovnik et al. [8] define the spoken attribute as the term that combines the binary and relative attributes to get better attribute definitions for images. Binary attributes are meaningful for describing one single image where relative attributes are significant for the comparison of two different images. In order to find out spoken attributes, ranking operation is conducted according to pairs of people in one single image. In this way, spoken attributes provide to compare people in a single image by merging the binary and relative attributes.

### 2.2. **Attribute Learning Methods**

**Binary Classifier Learning** Binary classifiers use the binary attributes to learn whether the image has attribute or not. For each attribute, learning classifiers that model the attributes are used and predict these attributes separately. Farhadi et al. [1] used three types of semantic attributes including shapes, parts and materials. They also used discriminative attributes to distinguish classes which have the same attributes. They used L1-regularized logistic regression to select effective features for classes. Kumar et al. [9] worked on two methods. First method consists of binary classifiers on attribute learning. There are sixty five attributes for face verification in their work and SVM classifiers are used for binary classification. Second method consists of simile classifiers which do not need label information and automatically find simile classes by using regions of faces. They found out if two images belong the same person by using two trait vectors of images with their extracted features. Firstly, binary classifiers are used and their results are saved in range of [-1,1]. Then, final SVM is used for separation boundary, to predicate similarity with the values close to zero. Lampert et al. [10] combined two types of predictions in their work. First one is direct attribute prediction (DAP) which is based on the attributes and makes decision from attribute comparison by

## Chapter 1. *BACKGROUND AND RELATED WORK*

using separate classifiers for each attribute. Second one is indirect attribute prediction (IAP) which is based on training classes and makes decision from training classes then attributes. In IAP, multiclass classifiers are estimated for each training classes. Jayaraman and Grauman [11] used different classifiers for each attribute to build decision trees for prediction.

**Weakly Supervised Learning** In this method training is applied via attributes of weakly labeled categories. Ferrari and Zisserman [12] performed this method with two attribute categories. First attribute category was extracted from one segment and used for colors that are red, green, blue, yellow. Second one was extracted from two segments that used for patterns like stripes, dots and checkerboard. Each image is represented by segments and image likelihood is calculated according to the model which consists of appearance, shape and layout features. Best background probability and best foreground segments are found by geometric properties. Models are learned separately and then the model which has best ratio is selected. They used the Google image search results for training and 20 percent of these images are not reliable which do not have the attribute that they want to learn. Rastegari et al. [13] used binary codes representations of images. They proposed two terms containing unsupervised similarity and discriminative similarity, that uses similar binary codes and different binary codes, respectively. To accomplish this, they used KNN and linear SVM for classification. This learning is based on categories and it does not guarantee that whether all images in category have the same binary codes.

**Ranking Based Learning** This learning method uses relative attributes. All images in a dataset can be compared with each other. To do this all attributes in dataset are ranked with ranking algorithms. Usually attribute comparison is made by three options with 'less', 'similar' or 'more'. Parikh and Grauman [7] worked on relative zero-shot learning. They learned classifiers by using ranks of each attribute. For the ranking algorithm they used Joachims's ranking SVM [14]. Then they ranked all images in the dataset by classifier results.

Relative zero-shot learning uses ranking based classifiers. Parikh and Grauman [7] worked on two types of learning. Firstly, they used direct attribute prediction (DAP) model of Lampert et al. [10] for zero shot learning. Relative zero-shot learning was the second type. They learned classifiers by using ranks for each attribute. When an image which belong unseen category came up they firstly found score of ranking for its attributes. Then they assigned

## Chapter 1. *BACKGROUND AND RELATED WORK*

image to unseen classes if it has the highest likelihood score is not available in seen categories.

This learning method is mostly used in image search. Attributes offer a similar structure to the keywords. For example, high-heel attribute for shoes is an important feature for someone who is looking for high heels shoes in product search. Image search consists two stages. Offline learning is the first stage and attribute classifiers are learned in this stage. Then, all images are ranked separately according to each attribute. In the online stage user gives a query to system and system finds the ranks according to the characteristics of the image of the query and, the system displays the most relevant ones. Vaquero et al. [15] worked on people search, which include some parts of people like upper face part, middle face part, lower face part, facial hair type, torso and legs. Firstly, they detected faces with AdaBoost classifiers and used nine Viola-Jones detectors to find facial attributes. They found torso and legs attributes by using color classifiers in HSL space.

Feris et al. [16] improved the previous work [15] and they added ranks of attributes to people search. Herewith they selected the most relevant images and showed to user. Kumar et al. [6] created a face tracer system that can search people by their faces. They divided faces into regions and used their regions' features for classification. Their classification method combines the SVM and Adaboost classifiers. Firstly, they used SVM classifiers for all attributes, then they used Adaboost to round SVM weights of attributes.

Relative feedback is one of the methods used for image search. It is quite difficult to find a good result with single keyword. Searches can be improved by taking feedbacks from users. User feedbacks are used for filtering results with each iteration. Kovashka et al. [2] worked on two types of feedback. First one is binary relevance feedback that user gives the feedback by 'relevant' or 'unrelevant' choices. System refines the results according to this selection by using SVM. Second one is relative attribute feedback and user has three choices (less, more, similar) in this type. Kovashka et al. ranked all attributes in dataset by using Joachims's SVMRank code [14]. When the user makes his choice, the attributes are ranked and their scores are updated by selected image. For example, sportive and also colorful shoes are wanted. The user selects an image from sportive shoes and indicates 'more' options with attribute 'colorful'. The system finds images which have high rank score by comparing the selected image.



## Chapter 1. *BACKGROUND AND RELATED WORK*

**Active Learning** Active learning requires a supervisor from outside the system. Suppose that we have a query image and we want to know whether this bus. Supervisor responds to this query, and says to us about why. For example, the reason of why the vehicle cannot be bus is may be the vehicle is too small to be a bus, and it must be larger. After that, the system accepts all images is smaller than this vehicle are not bus. In this manner system can learn information of images which are not labeled. In this way, the system can get more label information by using less effort. Parkash and Parikh [17] used this method. They applied binary classifier to each category by using RBF SVMs. They improved binary classifiers result by explanation of supervisors about the attributes of classes. The explanation is expected only if classifier gives wrong choice and the user can give one attribute explanation. System picks an image, which is not labeled and asks to supervisor according to the maximum result of binary classifiers. Supervisor gives an answer and if the image does not belong to the suggested class gives an explanation too. Then, the system update the models according to this explanation. They used Joachims's ranking SVM [14] for attribute ranking.

Different from regular supervised learning, in active learning, supervisors indicate reason for their labels. Then, all attributes are handled from these reasons. For example, users are asked which images more attractive. Users can say that some of images more attractive because they are more colorful. All subsequent images like having these colors will be automatically considered attractive. Kovashka and Grauman [18] initialized a graph for object attribute model. Then, they updated their labels for object and attributes by supervisor feedbacks with entropy-based selection function. If the object label changes by supervisor old label throws and new labeled. If the attribute label changes it is added to attributes of image. They used multi-class SVM for object classes by ignoring attributes. They used binary classifiers for attributes by ignoring object classes. Their model consists of object classes, attributes, attribute-attribute relationships and object-attribute relationships.

Biswas and Parikh [19] used active learning with attributes based feedback. A supervisor is expected to give a label feedback, which is true or false then give an attribute based explanation in case labeling is false. These explanations can be like 'too' or 'not enough' options. They used binary classifier for each category as SVMs. Weighting schemas are used in this work to consider all attributes in image and entropy of system are calculated to actively select an image. Liang and Grauman [20] used active learning to rank relative attributes because ranking of relative attributes is costly. In this work images are selected by pool-based approach which prediction of ranking group images benefits to learn. They used Joachims's

ranking SVM [14] with 1-D ordering features.

**Multi-Attribute Learning** Binary and ranking attribute learning models get one attribute and learn classifiers for only this attribute. Multiple attributes may be requested in applications. To do this, attributes can be learned separately then attributes can be combined according to the user request. Scheirer et al. [21] used different classifiers for each attribute and tried to calibrate of these SVM scores. Douze et al. [22] used Fisher vectors and SVMs for multi-attribute learning. They learned each attribute classifier with non-linear binary SVMs. Then they used L2 norm normalization on attribute scores and combined them with Fisher vectors. But these methods could be difficult to giving weights of all attributes. Instead of these methods multi-label queries can be used. Siddiquie et al. [23] combined retrieval and ranking to achieve multi-label attribute learning. In multi-label queries they handled attribute classifiers together instead of using separate classifiers for each attribute. They used bundle methods for regularized risk minimization and evaluated their works on facial datasets.

**Human in the loop recognition** Some recognition problems can be hard for people. For example, when defining a bird with what color of feathers are easy for people. But it is difficult to tell which bird species it belongs. In such cases, human in the loop recognition is used that human interaction helps the learning in runtime. User gives an input image to system and system attempts to identify the image with the feedback given by the user. If we talk over our example, user wants to find the species of a bird. System begins to ask questions via the supplied picture. The user would will have reached the correct species by the answers of him. Wah et al. [24] asked twenty questions to user. The user gave the results and also qualified his result by 'guessing', 'probably' and 'definitely' choices. They updated their models with user answers by using maximum information gain. They used one vs all SVMs for each classes. On validation set Platt scaling is used.

Branson et al. [25] approached this problem by combining part-based models and attribute learning. In attribute learning part they used separate classifiers for each attribute. In detection part they found detection scores of images with sliding window parts by using structured SVMs. User response the questions and gives qualification degree of his answers from 'guessing', 'probably' or 'definitely' choices. After each answer, probabilities are recalculated again by information gain. The system asks the question with the maximum information gain to the user and this iteration goes on until the result is found.

## Chapter 1. *BACKGROUND AND RELATED WORK*

**Deep Learning** In recent years Deep Convolutional Neural Networks (CNNs) [4] are frequently used in computer vision. This method is used mostly for object recognition, but some works used it for attribute learning. Shankar et al. [26] are worked on attribute learning with Deep Convolutional Neural Networks (CNNs). In their work for attribute learning pseudo-labels are used. Each fixed number of iterations the responses are analyzed and multiple attribute labels were handled. To do this, they used feature maps of images in convolutional layers. After that, they tried to learn attributes by using the average spatial response. They called deep-carving method to their work which analyses the features in the training stage and try to learn missing labels to get fine attribute-specific feature maps. Razavian et al. [27] worked on different tasks by deep learning and they handled attribute detection in their work by using OverFeat network. Their feature vector is the first fully connected layer of this network and its size 4096.

**Transfer Learning** Attributes are handled mostly like category-independent properties. But in some cases this generalization can be wrong. Grauman and Chen [28] worked on to create category-sensitive attribute models. They created some classifiers with labeled data and created analogous attribute classifiers with unlabeled data. For labeled data they used category-sensitive SVMs for each classes with their attributes whether presence. They factorized a tensor for each object and attribute to build latent structure. After that, they used K-dimensional latent feature vectors to find the same attributes how look in other categories.

**Multi-Task Learning** Some attributes can be sensitive to training images. Let's think some images in training set have wheel attribute and all images have metallic wheels. Learning in this situation could be wrong because metallic and wheel attributes are correlated. In test stage the new image prediction will fail because of the wooden wheel. To prevent this Hwang et al. [29] proposed multi-task learning for attribute learning. They wanted to decorrelate the attributes that are semantically distinct each other and, correlate attributes that they are sharing semantically similar. They used jointly classifiers instead of using separately classifiers. They used logistic regression classifier model with L2 normalized to find in-group sharing and logistic regression classifier model with L1 normalized to inter-group competition.

### **2.3. Tasks and Applications**

Dhar et al. [30] worked to predict aesthetics and interestingness of images. They used 3 types of attributes. Compositional attributes related to layout, content attributes related to consist of some objects and sky-illumination attributes related to illumination of image. Feris et al. [31] worked on searching vehicles in traffic videos. They used attributes to help searching keywords like 'show me yellow cars'.

Saleh et al. [32] found out abnormal images from set. They used qualitative and quantitative analysis for this work and they brought a point of view how people looking abnormalities.

Image search can be expensive because of it scan all dataset for one attribute. Kovashka et al. [33] tried to find a solution to this problem with attribute pivots. They used binary search trees for searching. Each attribute have some pivots. Searching by query begins with these pivots. Then the others images from these pivots are investigated by left or right children. Kovashka et al. [34] tried to implement user specific attribute models. They collected the labeled data according to explicitly ask to user for labeling. Then, they mined the user's search history to find labels implicitly.

In images, every attributes may not draw attention. Some attributes draw more attention than others. Turakhia and Parikh [35] tried to find out this attributes in images. Christie et al. [36] used attributes in their predictable annoyance work. They predicted the user annoyance when they react occurring mistakes in computer vision systems. They learned mistakes by using the example mistakes. Kovashka and Grauman [37] gave a new approach with shades attributes. These attributes are different interpretations of users. Variant of an attribute are learned by using this method. For example, open attribute for shoes can be learned 'open at heel' or 'open at toe'. In this way attribute learning can be more generic. Bourdev et al. [3] used part-based approach to learn attributes of people. They wanted to learn these attributes under the condition of the viewpoints and poses.

### **2.4. Low-level Features Used in Attribute Learning**

All low-level features can be used in attribute learning. Table 2.1 shows the low-level features that have been used in attribute learning methods that have been explained in the previous subsection.

## Chapter 1. *BACKGROUND AND RELATED WORK*

Color histograms are used to represent distribution of colors of image. In attribute learning methods mentioned above used RGB, CIELAB and HSV color spaces. Scale-Invariant Features (SIFT) [38] are used on gray-scale images and provide to robustness to image rotation, image scaling and image translation. Color SIFT models are used to like SIFT also adding to color information. HueSIFT, HSV-SIFT, OpponentSIFT, rgSIFT, CSIFT, RGBSIFT are the color SIFT models [39]. In the related works rgSIFT is mostly used. Haar-like features are calculated with the rectangular parts intensities and used for face detection [40]. Texture features are used to achieve patterns of an image. Some of the related work used this texture feature. Local Binary Patterns (LBP) [41] feature is like texture information, but it is calculated with neighborhood. GIST [42] descriptors are extracted by applying Gabor filters to image and usually are used in scene recognition. SURF [43] is used to detect interest points which are scale and rotate invariant.

According to Table 2.1, color histograms are the most frequently used low-level features for attribute learning. This is not surprising since color is an adjective and therefore attribute by definition, and color histograms are a straightforward way to represent them. A second observation from this table is that, shape features such as HOG [44] and SIFT [38] are also frequently used for attribute recognition. In addition, GIST [42] feature is also frequently used, probably aiming to capture the contribution of the global information to attribute recognition.

TABLE 2.1: Low-level Features commonly used in attribute learning

	Color Histograms	HOG	SIFT	ColorSIFT	Haar-like	LBP	GIST	SURF	Texture
Parikh and Grauman [7] 2011	X	-	-	-	-	-	X	-	-
Ferrari and Zisserman [12] 2007	X	-	-	-	-	-	-	-	X
Farhadi et al. [1] 2009	X	X	-	-	-	-	-	-	X
Kumar et al. [9] 2009	X	X	-	-	-	-	-	-	-
Jayaraman and Grauman [11] 2014	X	X	-	-	-	-	-	-	-
Lampert et al. [10] 2014	X	X	X	X	-	-	-	X	-
Parkash et al. [17] 2012	X	-	-	-	-	-	X	-	-
Kovashka et al. [18] 2011	X	X	-	X	-	-	-	-	-
Biswas and Parikh [19] 2013	X	-	-	-	-	-	X	-	-
Liang and Grauman [20] 2014	X	-	-	-	-	-	X	-	-
Vaquero et al. [15] 2009	X	-	-	-	X	-	-	-	-
Kumar et al. [6] 2008	X	-	-	-	X	-	-	-	-
Siddiquie et al. [23] 2011	X	-	X	-	-	X	-	-	X
Kovashka et al. [2] 2012	X	-	-	-	-	-	X	-	-
Wah et al. [24] 2011	X	-	X	-	-	-	-	-	-
Branson et al. [25] 2010	X	-	X	-	-	-	-	-	-

## 3. APPROACH

In this chapter, we first describe the low-level features and mid-level features that have been evaluated within the attribute learning framework. Then, we describe the attribute learning method we use. We also briefly describe how different features are combined in our evaluation framework.

### 3.1. Feature Types

In order to analyze the performance of the low-level features on attribute recognition, we chose four basic types of low-level features to evaluate. These are a) Color features, b) Shape features, c) Texture features and d) Hybrid features. In addition to low-level features, we also evaluate the recently evolving mid-level feature, which can also be referred as a semantic feature, Convolutional Neural Network (CNN) feature. Below, we describe each of these features in further detail.

#### 3.1.1. Color Features

As it is seen in table 2.1 all works use the color histograms that basic feature of attribute learning. In order to evaluate the effectiveness of color histograms, we used three different types of color histograms which use three different color spaces. These are RGB, HSV and CIE LAB color spaces, respectively.

**RGB Color Space** Primary colors red for 'R', green for 'G' and blue for 'B' are the coordinates of this color space. Each pixel are calculated by the range of 0 to 255 values of these colors.

**HSV Color Space** HSV is represented by cylindrical coordinates. 'H' is hue, 'S' is saturation and 'V' is the value of brightness. The hue is angular value of colors that value of red 0 to 120 degree, value of green 120 to 240 degree and value of blue 240 to 360 degree. Saturation and brightness values are between 0 and 1.

## Chapter 3. *APPROACH*

**CIELAB Color Space** There are three coordinates in this color space. 'a', 'b' are the color-opponent dimensions and 'L' is the lightness. 'L' is the lightness of colors that L=0 is black and L=100 is white. 'a' is the color between magenta and green that negative values define green and positive values define magenta. 'b' is another color between yellow and blue that negative values define blue and positive values define yellow.

Color histograms represent the distribution of colors in images. The color spaces are used to construct these histograms. Each pixel is handled one by one and the color value of the pixel is extracted according to color space. The distribution of all pixels gives the color histogram of image.

### 3.1.2. Shape Features

**Histogram of Gradients (HOG) :** Dalal and Triggs [44] represented the images by intensity gradients or edge directions. In their work, they used these features for human detection. Image convolution by filter with using the kernel  $[-1 \ 0 \ 1]$  is the first step to get HOG features. The second step is to find gradient magnitude and direction from regions of image which size 64x128 pixels. The third step is computing histograms of selected small parts of regions named by cells and usually have size 8x8 pixels. The next step is grouping cells into descriptor blocks with overlapping. This descriptor blocks can be circular or rectangular. The last step is normalization and L2-Hys, L2-norm and L1-sqrt, L1-norm normalization schemes are applied to this work and except L1-norm all other schemes performed results equally.

Dalal and Triggs [44] method was used in this work to get HOG feature. We used 8x8 block size and 4 pixel to step size using feature pyramid.

**Scale-Invariant Features (SIFT) :** Lowe [38] found a feature type, which is invariant to image scaling, image rotation and image translation on gray scale images. They used key locations that they are difference points after Gaussian functions applied to image. These points are maxima and minima according to its eight neighbors. But all of the key points are not meaningful in this stage. So Taylor series extrema value is used to refine the key points which have bad contrast. Other filtration is done on the edges that response only one direction. An image pyramid occurs with key points. Then, image gradients and orientations



## Chapter 3. *APPROACH*

are found with each level of pyramid. 16x16 pixels of gradients are calculated and they are divided into 4x4 regions. From each region eight image gradients and orientations are extracted. The result 4x4=16 size of histograms with one and all eight bins is 128 dimensional feature vectors.

In this thesis, we used the standard SIFT descriptors as described above.

### 3.1.3. Texture Features

**Texton** Textures are the patterns in images. Varma and Zisserman [45] used texton representation from textures. Texture descriptors are extracted from randomly chosen training images. Then, filter bank is applied to all texture descriptors. K-means clusters are done by using this filter responses. From these clusters the texton dictionary is formed. Training images are filtered by filter bank and the distance is calculated to texton dictionary for each response from these filters and closest response is picked for representation and to forming the model. The test image is represented with histogram by using texton dictionary. After that nearest neighbour classifier apply to find closest model. The four filter sets are applied to this model of learning which are The Leung-Malik(LM), The Schmid(S) and The Maximum Response(MR4, MR8). In their experiments MR8 filters gave the best result.

In this work texton method was applied the same way Varma and Zisserman [45] and MR8 filters which gave the best results in their work are applied to images.

**Local Binary Patterns (LBP) :** Ojala et al. [41] used texture unit which represents the texture spectrum of image, to create local binary patterns from gray scale images. Texture unit is calculated with eight pixels which in the neighborhood. The first step is to pick one center pixel and get 3x3 pixel neighborhood values. Then, difference between neighborhood and center pixel is found. If the difference is negative, then set zero else set one. So the center pixel is threshold value. Then, all values are multiplied by their weights and the sum of these values gives the LBP value. Ojala et al. [46] improved local binary patterns to rotation invariant. For achieve this purpose they consider the signs of neighborhood instead of difference value. All possible values are calculated for neighborhoods in case rotation.

We used 8 pixel cell size for neighborhood to compute LBP in this work.

### 3.1.4. Hybrid Features

**Color Invariant Characteristic SIFT Feature (CSIFT)** SIFT may not enough to represent an image because it is worked with gray scale images and color information is ignored. Whereas color information can be distinguishing feature. Color SIFT is created for this missing property of SIFT. There are some color sift models like HueSIFT, HSV-SIFT, OpponentSIFT, rgSIFT, CSIFT, RGBSIFT. In this work Abdel-Hakim and Farag [47] color sift feature representation which named CSIFT is used. They used the same strategy for SIFT to geometric invariant. The only difference is using color invariant gradients instead of gray gradients. RGB color space and Gaussian color models are used to giving input parameters to Kubelka-Munk theory that diffuse the reflectance for color invariants.

### 3.1.5. Deep Features

**Convolutional Neural Networks (CNN) Features** In multi-layer neural networks there are some neurons which get the inputs and do some operations and produce the outputs. The outputs of neurons can be the inputs of another neurons. This passing is called forward pass. The important part of the multi-layer networks is backpropagation. Backpropagation uses the loss function that gives the consistency of the work with comparing the ground truth labels with predicted labels. Backpropagation begins to operate with the last output and recursively calculate the loss function for all neurons to start. Backpropagation can be used more effectively by divided forward pass operation into the parts.

Convolutional neural networks are similar to multi-layers neural networks. In multi-layer neural networks, neurons have one dimension and it is not effective to images due to their three-dimensional structure. For example, if the image size is 50x60X3 it will be 900 weights passing as outputs and this will decrease the performance. Krizhevsky et al. [4] used convolutional neural networks with three dimensions layers, which is using for communicating and storing. Width, height and depth are the three dimensions. There are three main layers in this neural networks. These are convolutional layer, pooling layer and fully-connected layer. Pooling layers does not pass parameters so the gradient of error does not calculate for them. Gradient of error is calculated for convolutional and fully-connected layers.

**Convolutional Layer (CONV):** The connection of activation neurons are provided by local regions. The size of these regions is the one of hyperparameter and is called receptive fields.

### Chapter 3. APPROACH

This layer applies the filters to these local regions by using the filter number supplied by hyperparameter. The receptive field size, stride and the amount of zero padding are the hyperparameters for this layer and the output size is calculated by using them. This layer introduces an output that has square of receptive field size and image depth and number of filters. Backpropagation of this layer also is a convolutional layer.

Pooling Layer (POOL): Usually this layer is used after convolutional layers. For each depth slice it calculates the maximum activation and it provides reduce the parameter size and reduce the local region size. The spatial size of resizing and stride are the hyperparameters of this layer. Backpropagation is used with the tracking of maximum activation neuron value for calculation of gradient.

Fully-connected layer (FC): Local regions are fully pairwise connected to activation neurons in this layer. The size of this layer is 4096 and, the last part convolutional neural networks is after that and has size as 1000 and is used for compute class scores. Fully-connected layers can be considered convolutional layer by getting the filter size as 4096.

In addition to these layers Rectified Linear Units (ReLUs) are used for output of convolution layers and fully connected layer. This provides less time to training.

The most used convolutional neural network pattern is like the following:

INPUT  $\rightarrow$   $[[\text{CONV} \rightarrow \text{ReLU}] * N \rightarrow \text{POOL?}] * M \rightarrow [\text{FC} \rightarrow \text{ReLU}] * K \rightarrow \text{FC}$   
( $N \leq 3, M \geq 0, 0 \leq K < 3$ )

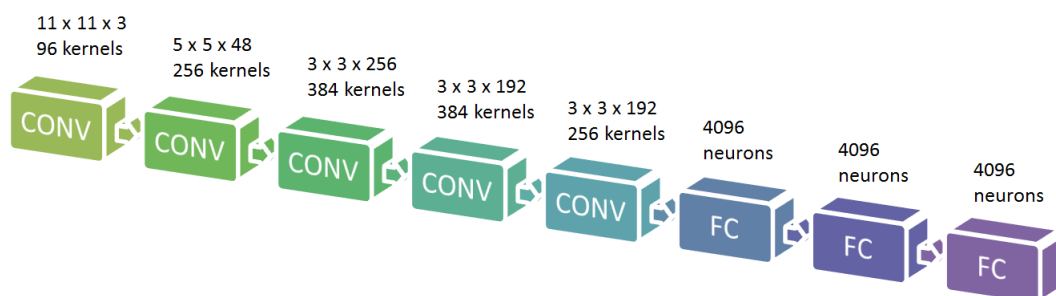


FIGURE 3.1.: Layers with kernel size of VGGNet Architecture

## Chapter 3. *APPROACH*

We used the output of fully-connected layer which size is 4096 as CNN feature in our work. There are some architectures for convolutional neural networks. We used VGGNet architecture [48] in this work that contains 16 convolutional/fully connected layers. The figure 3.1. shows the kernel size of layers. We used 'imagenet-vgg-verydeep-16' pre-trained models for extracting CNN features as in [48].

We should note that main point of this thesis is analyzing the effects of low-level features. CNN features are actually supervised features that can be considered as mid-level features. Since they achieve the current state-of-the-art in many recognition tasks, we aim to find out the effects of this representation to attribute recognition, and hence, we also include these features to our evaluation framework, and test their recognition performance both individually and in combination with low-level features.

### **3.2. Method**

The purpose of this work is to find out which low-level features are useful to predict attributes. Attributes can be different from each other and the low-level features that perform the best in recognizing each of them can be different. In our evaluation framework, to assess which low-level feature is the best for attribute learning, we use an attribute learning method, with different underlying features. For this purpose, we adopt the method of Farhadi et al. [1]. In their work, the main usage of attributes is to find out object classes. In this thesis, we omit the object recognition part since as noted above, attributes could be used in many different application domains and we focus on only attribute prediction since its performance is likely to affect corresponding tasks such as object recognition.

The overall attribute learning process is shown in Figure 3.2. and the main steps of this framework can be listed as follows:

- Low-level Feature Extraction
- Feature Selection
- Classifier Training and Prediction

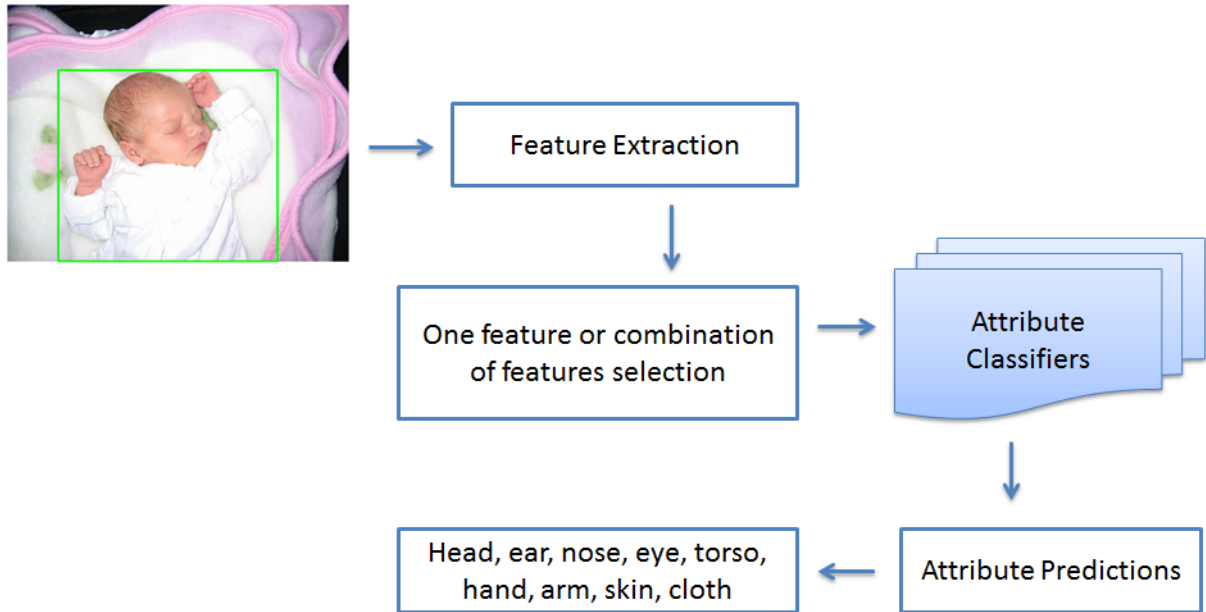


FIGURE 3.2.: Approach of this work

**Low-level Feature Extraction** In [1] three types of features are used. Base features consist of HOG, texton and CIELAB color histograms. We used these base features too and added some base features like SIFT, LBP, CSIFT, RGB color histograms, HSV color histograms. We used CNN as mid-level feature.

**Feature Selection** Attribute classifiers could fail if they learn with correlated attributes. For example, we want to learn 'furniture arm' classifiers and in our dataset all furniture has wooden arms and 'wooden' is another attribute in this dataset. Learning process is constructed with these training images and 'furniture arm' classifiers are affected with 'wooden' attribute. When the new furniture is added to dataset with plastic arms the learning will fail. These attributes cannot be separable in this situation. To resolve this problem learning classifiers are designed with or without the attribute for object recognition. For example, we want to learn 'furniture arm' classifier then, we calculate the result of 'sofa' object recognition with and without 'furniture arm'. The confusion of classifier between the 'furniture arm' and 'wooden' attribute can be solved by implementing this method. Features are selected to represent most important features to learn classifiers by applying L1-regularized logistic regression. This regression provides the best features by trying different parameters which are dependent attribute result according to nearest ground truth values. The features of other objects which has the 'furniture arm' attribute are extracted by the same L1-regularized logistic

### Chapter 3. APPROACH

regression. Then, all of these features are pooled to learn 'furniture arm' classifier. In our example 'armchair' object class has 'furniture arm' attribute and the features are extracted by using L1-regularized logistic regression to distinguish with or without this attribute. Then, we pool the features from 'armchair' and 'sofa' to learn 'furniture arm' classifier. In [1], selected features as mentioned above and whole features are compared and the correlation of the attributes by using selected features gave lower rate. As a result, it can be said that selected features are less sensitive to biases in dataset.

**Classifier Training and Prediction** Learning a model is the first step of classification. From extracted features in training images we learn a model then classify them by using this model. We used L1 regularized logistic regression to create models. Linear regression is the base of regression. This type of regression is used to get binary results. We can select an example which is based on getting probability sales of water according to temperature. The probability should be between 0 and 1. Linear regression gives us a model from training temperatures with formula in 1.

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \varepsilon \quad (1)$$

The model does not support very high or very low temperature because it cannot be negative or greater than 1 which is contradictory with probability. This regression type will fail for these reasons and we will have underfitting problem which the training data does not fit in model, and another problem will occur that called high bias. To prevent these problems, logistic regression is used. Logistic regression can be considered as probability of linear regression with formula in 2.

$$P(y|x) = \frac{1}{1 + e^{-yw^T x}}, \quad \text{where } y = \pm 1 \quad (2)$$

Using these high polynomial terms cause the overfitting problem that fit the training data well, but incapable of learning new data, and another problem comes with high variance. To avoid these problems there are two methods. The size of features can be reduced or regularization can be applied. The regularization method keeps all features, but changes

## Chapter 3. APPROACH

their weights with formula in 3.

$$\min_w \|w\|_1 + C \sum_{i=1}^l \log(1 + e^{-y_i w^T x_i}) \quad (3)$$

$\|\cdot\|$  as 1 - norm

The parameter selection for regularization is essential. We want to fit the model on train well and get the parameter which gives the weights smaller. Loss function is used to learn the best parameter which obtain the result of loss function with lowest value. L1-regularized logistic regression reduces the parameter and encourages the sparsity in order to find the most important features. In this way less important features are eliminated with value of zero and most important ones are selected.

In this work all attributes have the separated classifiers. After extracting the features, attribute classifiers are learned. Then, attribute predictions are found for each classifier.

**Feature Combination** In this thesis, we also look at the performance of the feature combinations, as well as the performance of the individual features. For this purpose, we utilize three types of feature combinations, which are:

- **Early Fusion** In this type of fusion, the feature vectors are extracted as described above and these vectors are concatenated consecutively. The combinations are composed by early fusion in this work unless otherwise indicated.
- **Late Fusion** In late fusion, prediction scores coming from the individual feature classifiers are combined. This type of fusion, each classifier response is multiplied with a weight  $w_i$  where  $\sum_i w_i = 1$  and the sum of the weighted scores is taken as the final prediction score. Here, when combining scores, we give equal weights to each of the classifiers. We used late fusion to evaluate top 4 features combinations with the best feature.
- **Weighted Late Fusion** In weighted late fusion, the prediction scores coming from the individual feature classifiers are used as in late fusion. For each model weights are computed with cross validation method that is used on only train images and obtain a validation set to find accuracy of models. We used the best accuracy from the cross validation results for weights of classifiers. In weighted late fusion, each classifier

weight is computed according to equation 4 as taken  $z_i, \dots, z_n$  best accuracy for each attribute,  $s_i, \dots, s_n$  score for each attribute.

$$w_i = \frac{\sum_{i=1}^n z_i \cdot s_i}{\sum_{i=1}^n z_i} \quad (4)$$

### 3.3. Implementation Details

In this part we explain the usage of features.

- **Color Features** We used kmeans cluster method on the color histograms. Visual vocabulary computed with 128 kmeans centers. Nearest codevector index was used for building color histograms. Color histograms were densely sampled.
- **Shape Features** HOG features were constructed with spatial pyramid. 8x8 blocks, two scale factor, four pixel step size were used to construct pyramid. HOG descriptors were used also with kmeans clustering. Feature histograms were formed with the nearest 1000 kmeans centers. HOG extraction were done by using the source code as handled in [1].

SIFT descriptors were extracted with the Vedaldi et al. [49] library. After extracting SIFT descriptors, kmeans clustering was performed and kmeans centers were extracted. SIFT descriptors were quantized to nearest 1000 kmeans centers. These descriptors were handled with the coordinates of the frames.

- **Texture Features** In this work texture features were extracted and The Maximum Response(MR8) filter set and Gaussian filter were applied. The texton filter bank is constructed. Textons were used with kmeans clustering method. The features are quantized to nearest 256 kmeans centers. Textons were extracted by using the source code as handled in [1].

By using the [49] library we extracted the LBP features too. The cell size was required for LBP which is using for neighborhood limit. We chose eight for this parameter. Kmeans clustering was used for this feature. The LBP features were quantized to nearest 1000 kmeans centers.



### Chapter 3. *APPROACH*

- **Hybrid Feature** We extracted the CSIFT descriptors by using ColorDescriptor software v4.0, which is created by Sande et al. [50] [39]. CSIFT descriptors were densely sampled at every six pixels. Kmeans clustering was used for CSIFT descriptors and 1000 kmeans centers were formed. The CSIFT descriptors were quantized to nearest 1000 kmeans centers. The output of this descriptor includes the coordinates. So, we used this information when for building CSIFT histograms.
- **Semantic Features** We used CNN features from mid-level features. We extracted CNN features by using library of Vedaldi and Lenc [48]. The fully connected layers output of the last layer was extracted and used as a CNN feature. This feature size was 4096.

For all features mentioned above if the dataset has bounding box information, the histograms of features are calculated with each bounding box bins. The dataset without bounding box information is handled with the full-size of image and its histograms.

## 4. EXPERIMENTS & RESULTS

In this chapter, we present the experimental evaluations of using various low-level features for the problem of attribute recognition. First, we describe the datasets used in our evaluations. Then, we give detailed experimental results and related discussions.

### 4.1. Datasets

**a-Pascal Dataset** Farhadi et al. [1] used The Pascal VOC 2008 dataset for object recognition and they extended this dataset with attribute annotations. Object classes in this dataset can be grouped by person, animals, vehicles, and indoor. In animals group there are sheep, bird, dog, cat, horse, and cow object classes. In vehicles group there are bus, aeroplane, motorbike, bicycle, car, train, and boat object classes. In indoor group dining table, bottle, sofa, chair, tv/monitor and potted plant object classes. There are 20 object categories in total and each category has 150 to 1000 objects. Sixty four attributes are defined to represent these classes. Each label for attributes and object classes are gotten from Amazon's Mechanical Turk. This dataset has the object bounding box information. The size of images can be different from each other. This dataset is separated to training and test images. In training there are 2113 images and in testing there are 2227 images. Figure 4.1. shows example images of this dataset.

**a-Yahoo Dataset** a-Yahoo dataset created by Farhadi et al. [1] for object recognition. It is collected by Yahoo image search. There are twelve object categories which are similar to a-Pascal dataset according to labels and bounding box information. But this dataset has different object categories and attribute correlations from the a-Pascal dataset. The object classes are building, donkey, wolf, statue of people, zebra, goat, monkey, bag, jet ski, carriage, mug and centaur. This dataset also have label of attributes like a-Pascal dataset and 64 attributes are defined. Amazon's Mechanical Turk users labeled the attributes and object classes in dataset. This dataset also has the information of bounding box, images can have more objects in each image. The size of images can be different. There are 2644 images in this dataset. Figure 4.2. shows example images of this dataset.

Chapter 4. *EXPERIMENTS & RESULTS*

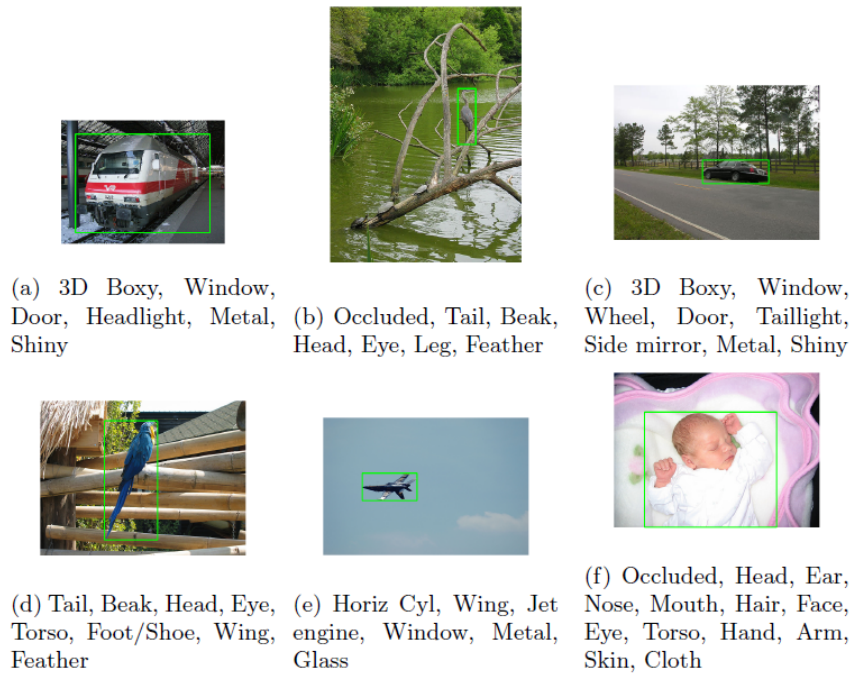


FIGURE 4.1.: Example images and their corresponding attributes for a-Pascal Dataset [1]

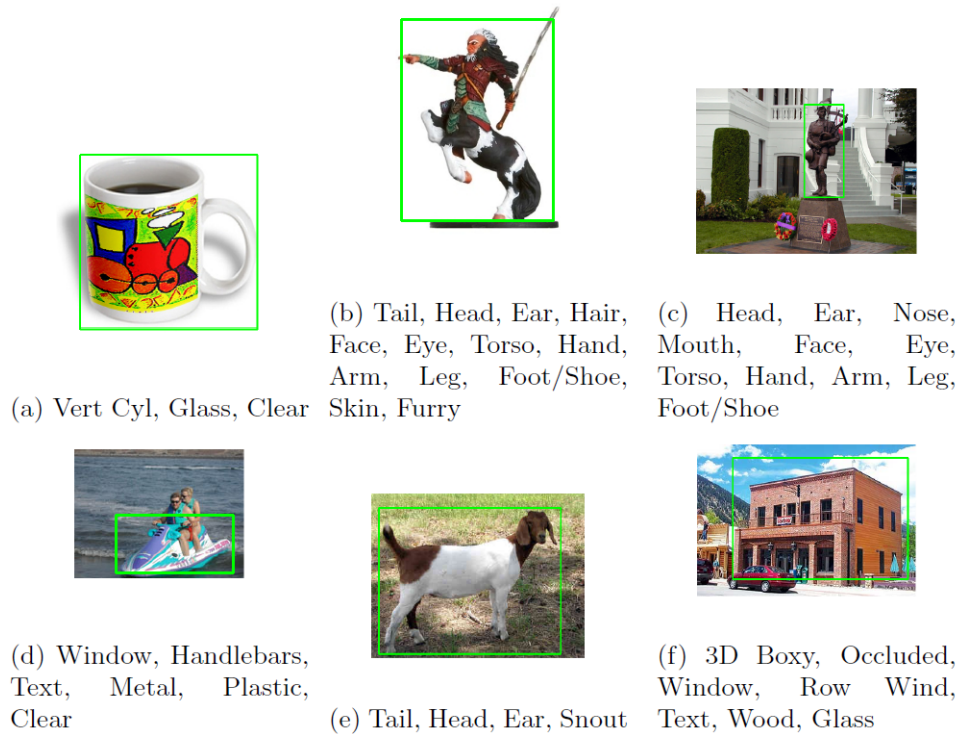


FIGURE 4.2.: Example images and their corresponding attributes for a-Yahoo Dataset [1]

## Chapter 4. *EXPERIMENTS & RESULTS*



FIGURE 4.3.: Example images and their corresponding attributes for Shoes Dataset [2]

**Shoes Dataset** Berg et al. [51] created the Attribute Discovery Dataset that has four shopping categories which are bags, earrings, ties, and shoes. Kovashka et al. [2] used the shoes dataset from these shopping categories. Amazon’s Mechanical Turk users labeled the dataset with binary and relative attributes. We used only binary attribute labels for our work and only shoes dataset. There are 14,658 images in dataset and the images have the same 280x280 pixels size. There is no bounding box information because there is only one object in each image. There are 10 attributes in total which are ‘pointy at the front’, ‘high at the heel’, ‘covered with ornaments’, ‘bright in color’, ‘open’, ‘long on the leg’, ‘feminine’, ‘sporty’, ‘shiny’, and ‘formal’. Figure 4.3. shows example images of this dataset.

**Attributes of People Dataset** Bourdev et al. [3] created a dataset that used for people attribute recognition. There is bounding box information for each image. All dataset are labelled by nine attributes. They are ‘is male’, ‘has jeans’, ‘has long hair’, ‘has t-shirt’, ‘has hat’, ‘has long sleeves’, ‘has shorts’, ‘has long pants’, and ‘has glasses’ properties of people. There are 4013 training and 4022 testing images and all images in dataset have different size. Figure 4.3. shows example images of this dataset.



FIGURE 4.4.: Example images and their corresponding attributes for Attributes of People Dataset [3]

## 4.2. Experiment Implementation Details

In our experiment we used four dataset as mentioned above. Attribute classifiers are learned for each attribute in each dataset. Fan et al. [52] created a LIBLINEAR library that use regression types. LIBLINEAR provides the multi-class classification with L1-regularized logistic regression. In the learning stage, we select the best C parameter by using cross-validation which divide the images by training and testing and try to find the best accuracy. The cross-validation provides learning better classifier that accurately predict unknown testing data. The overfitting problem can also be prevented by using cross-validation.

## 4.3. Experimental Results

In the evaluation part we separate the train and test classes as follows. We used a-Pascal train set according to the [1] for learning and a-Pascal test set in reference to [1] for testing. When using a-Yahoo dataset at the learning stage we used a-Pascal models and tested them with

a-Yahoo dataset. The Shoes dataset does not have the information of train and test categories. Train and test sets ratio of other datasets, we divided the dataset by two and used for train and test images. For Attributes of People dataset the train and test sets are prearranged so we used them for training and testing.

**Performance of Individual Features** This part is separated into color features and other features that used as low-level features.

- **Color Features** We used color histograms with three color spaces. RGB, LAB and HSV color spaces were evaluated in this work. LAB color histograms gave the best results for a-Pascal and Attributes of People datasets in reference to tables 4.1 and 4.4, respectively. RGB color histograms mostly gave the best results for a-Yahoo dataset as it is shown in the table 4.2. For Shoes dataset the HSV color histograms gave the best results in reference to Table 4.3.

Features	ROC Area	AP
RGB Color	0.760	0.298
HSV Color	0.768	0.311
LAB Color	<b>0.771</b>	0.293

TABLE 4.1: The results of color features for a-Pascal dataset

Features	ROC Area	AP
RGB Color	<b>0.668</b>	0.198
HSV Color	0.667	0.193
LAB Color	0.659	0.187

TABLE 4.2: The results of color features for a-Yahoo dataset

Features	ROC Area	AP
RGB Color	0.898	0.871
HSV Color	<b>0.903</b>	0.877
LAB Color	0.893	0.867

TABLE 4.3: The results of color features for Shoes dataset

Features	ROC Area	AP
RGB Color	0.623	0.281
HSV Color	0.616	0.277
LAB Color	<b>0.630</b>	0.283

TABLE 4.4: The results of color features for Attributes of People

- **Shape Features** We used HOG and SIFT feature types in this category. HOG is the best shape feature among them. For all dataset HOG features gave the best result as shown in tables 4.10, 4.11, 4.12, 4.13. So, if user wants to use shape feature in attribute learning, he should use HOG feature.

We wanted to learn effects of kmeans center count on the results. The tables 4.5 and 4.6 show the results of different kmeans center count. For a-Pascal and a-Yahoo datasets the lowest value resulted in 250 kmeans centers and 750 kmeans centers provide the best result. We can analyze these tables 4.5 and 4.6 as if we used small count of centers it will learn small vocabulary and it will be inadequate to represent visual words, and if we used large count of centers it will learn big vocabulary and visual words will be pointless parts.

Features	ROC Area
250 Centers HOG	0.859
500 Centers HOG	0.874
750 Centers HOG	<b>0.876</b>
1000 Centers HOG	0.870

TABLE 4.5: The results of different k-means center clustering of HOG feature for a-Pascal dataset

Features	ROC Area
250 Centers HOG	0.820
500 Centers HOG	0.832
750 Centers HOG	<b>0.832</b>
1000 Centers HOG	0.828

TABLE 4.6: The results of different k-means center clustering of HOG feature for a-Yahoo dataset

- **Texture Features** We used texton for texture features. This feature worked better from color histograms in Shoes and a-Yahoo dataset as shown in table 4.11 and 4.12 and worked worse in a-Pascal and Attributes of People datasets as shown in 4.10 and 4.13. Attribute learning by using only this feature may not give the good results. We

Chapter 4. *EXPERIMENTS & RESULTS*

also used LBP features for texture feature. In between SIFT and LBP features, SIFT better performed in a-Pascal and Shoes dataset and LBP is better performed in a-Yahoo and Attributes of People dataset.

- **Hybrid Feature** In hybrid category we used the CSIFT descriptors. CSIFT descriptors gave better result as compared with SIFT features, also CSIFT descriptors have better performance as compared with color histograms. So using color invariant in SIFT features improved results of SIFT feature in attribute learning. We wanted to analyze the effects of kmeans center count on CSIFT feature. The tables 4.7, 4.8 and 4.9 show that kmeans center count vary in accordance with datasets. In addition to this the results have one thing in common that they do not give good results for very small count of centers.

Features	ROC Area
250 Centers CSIFT	0.959
500 Centers CSIFT	<b>0.964</b>
750 Centers CSIFT	0.962
1000 Centers CSIFT	0.955

TABLE 4.7: The results of different k-means center clustering of CSIFT feature for Shoes dataset

Features	ROC Area
250 Centers CSIFT	0.856
500 Centers CSIFT	0.865
750 Centers CSIFT	<b>0.875</b>
1000 Centers CSIFT	0.874

TABLE 4.8: The results of different k-means center clustering of CSIFT feature for a-Pascal dataset

Features	ROC Area
250 Centers CSIFT	0.792
500 Centers CSIFT	0.807
750 Centers CSIFT	0.808
1000 Centers CSIFT	<b>0.817</b>

TABLE 4.9: The results of different k-means center clustering of CSIFT feature for a-Yahoo dataset

- **Semantic Features** We used CNN features in this category. Except a-Yahoo dataset CNN is the best feature for attribute learning according to the results as shown in Tables 4.10, 4.11, 4.12, 4.13.



Chapter 4. *EXPERIMENTS & RESULTS*

<b>Features</b>	<b>ROC Area</b>	<b>AP</b>
LBP	0.731	0.252
SIFT	0.734	0.340
Texton	0.758	0.265
LAB Color	0.771	0.293
HOG	0.871	0.497
CSIFT	0.874	0.505
CSIFT*	0.875	0.507
HOG*	0.876	0.511
CNN	0.878	0.473

TABLE 4.10: The results of low-level and mid-level features for a-Pascal dataset  
\* : 750 kmeans centers

<b>Features</b>	<b>ROC Area</b>	<b>AP</b>
RGB Color	0.668	0.198
Texton	0.719	0.234
SIFT	0.731	0.333
LBP	0.748	0.314
CNN	0.814	0.445
CSIFT	0.817	0.381
HOG	0.828	0.427

TABLE 4.11: The results of low-level and mid-level features for a-Yahoo dataset

<b>Features</b>	<b>ROC Area</b>	<b>AP</b>
LBP	0.895	0.871
HSV Color	0.903	0.877
Texton	0.924	0.910
SIFT	0.947	0.936
CSIFT	0.955	0.943
HOG	0.968	0.961
CNN	0.983	0.979

TABLE 4.12: The results of low-level and mid-level features for Shoes dataset

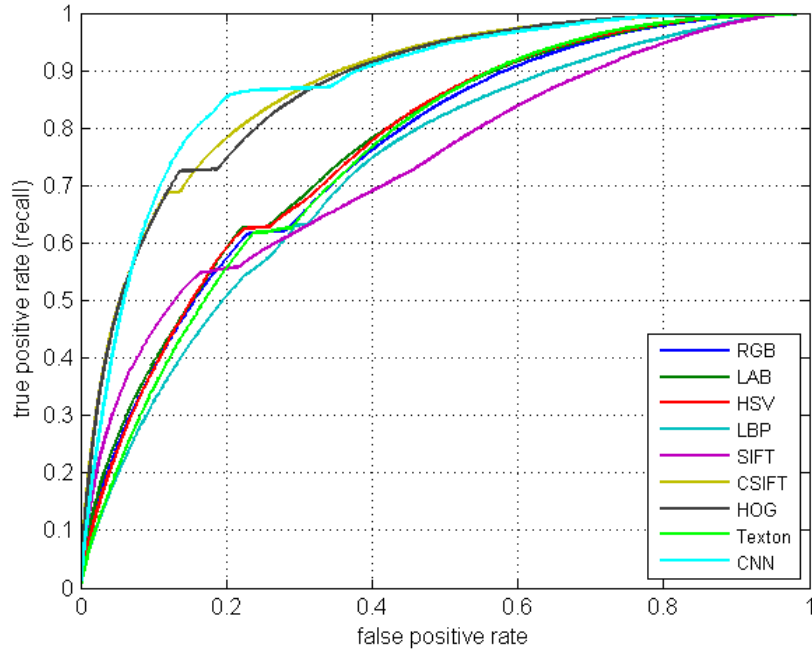


FIGURE 4.5.: ROC curves of low-level and mid-level features for a-Pascal dataset

Features	ROC Area	AP
Texton	0.554	0.257
SIFT	0.576	0.266
LBP	0.607	0.242
LAB Color	0.630	0.283
HOG	0.688	0.337
CSIFT	0.726	0.384
CNN	0.805	0.498

TABLE 4.13: The results of low-level and mid-level features and their combinations for Attributes of People dataset

According to our experimental results on evaluating individual features in Tables 4.10, 4.11, 4.12, 4.13, CNN features can be regarded as the most effective features for attribute learning, except for the a-Yahoo dataset as shown in all ROC curves for datasets in Figures 4.5., 4.6., 4.7., 4.8.. This result is not surprising, since CNN features are supervised features that have been extensively trained using additional data. CSIFT feature is the second-best feature in a-Pascal and Attributes of People Dataset. HOG feature is the second-best for Shoes dataset. The results of a-Yahoo dataset are different from other datasets. For a-Yahoo dataset HOG feature is the best and second-best is CSIFT feature.

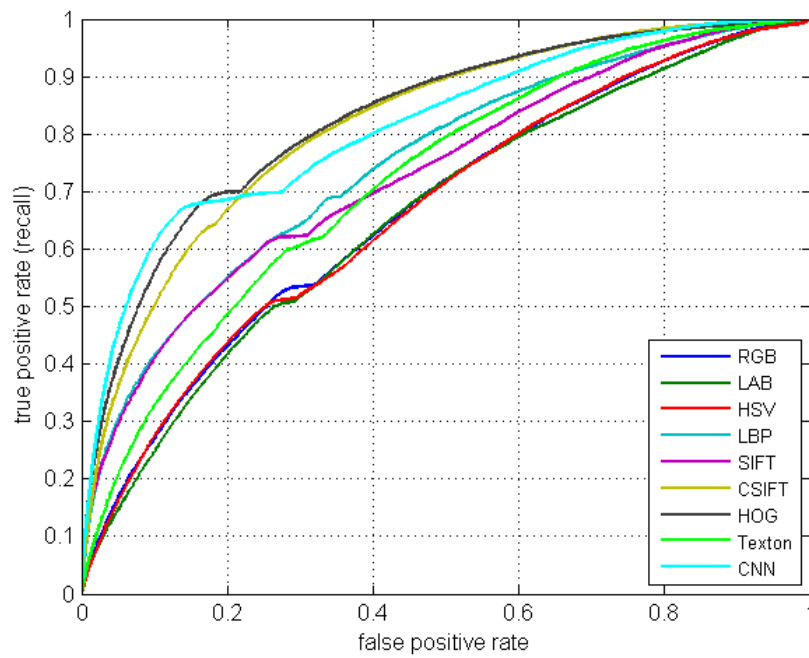


FIGURE 4.6.: ROC curves of low-level and mid-level features for a-Yahoo dataset

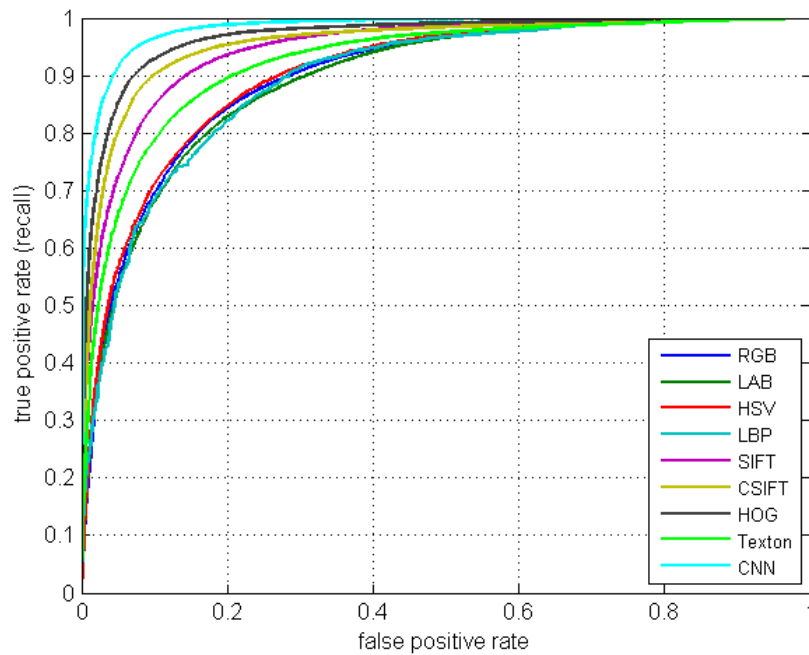


FIGURE 4.7.: ROC curves of low-level and mid-level features for Shoes dataset

We got the results of each attribute separately and we concatenated them, finally we created ROC curves for combined attribute scores. Apart from those, we used the result of each attribute separately and demonstrated them in figures for a-Pascal dataset 4.9., 4.10., 4.11., 4.12., 4.13., 4.14., 4.15., for a-Yahoo dataset 4.16., 4.17., 4.18., 4.19., 4.20., for Shoes dataset 4.21. and for Attributes of People dataset 4.22.. The purpose of these charts is to present the correlation of attribute and low-level feature types.

In a-Pascal dataset the body parts like ear, nose, mouth, hair, face, eye, torso, hand, arm, leg, gave the best results with HOG feature as shown in figures 4.10. and 4.11.. CSIFT descriptors are the second-best and CNN is the third-best for these types of attributes. The object parts like furniture arm, furniture leg, furniture seat, furniture back, rein, saddle, propeller, jet engine, window, row wind, wheel, pedal, handlebars, sail, engine, mast, gave the best results with CNN feature as shown in figures 4.11., 4.12., 4.13., 4.14.. The HOG feature is mostly the second-best for these attributes and CSIFT is mostly the third-best feature for this type of attributes. In the small part of these attributes HOG feature and CSIFT replace each other. The material attributes like leather, wool, feather, wood, plastic, gave the best results with CNN feature as shown in 4.14. and 4.15.. The second-best and third best replaceable with

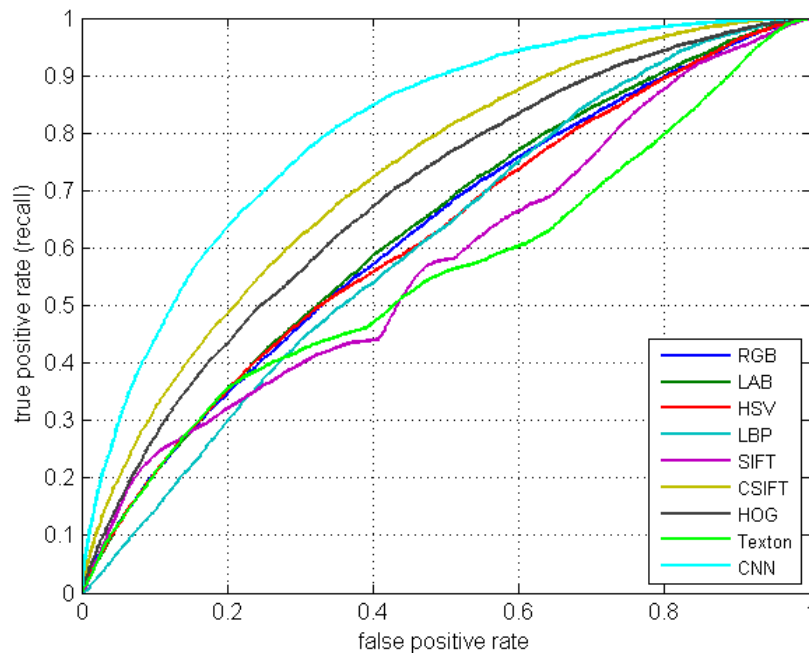


FIGURE 4.8.: ROC curves of low-level and mid-level features for Attributes of People dataset

HOG and CSIFT features. Some of them like metal gave the best results with CSIFT feature. The spatial attributes like 2D boxy, 3D boxy, vertical cylinder, horizontal cylinder gave the best results mostly with CNN features. CSIFT and HOG results for these attributes are very close together as shown in figure 4.9.. The color histograms gave varying results regarding to attributes.

In a-Yahoo dataset these results changed. The reason can be explained with the data distribution of two different dataset. The figures 4.16., 4.17., 4.18., 4.19., 4.20. show the results of a-Yahoo dataset. There are some missing attributes in a-Yahoo dataset that are defined for a-Pascal dataset. We did not add these attributes in figures. Similar to a-Pascal dataset this dataset also gave the best result for body parts like hand, arm, face, eye, head, ear, nose with HOG features as shown in figures 4.16. and 4.17.. Some material attributes like metal, leather, plastic gave the best results with CNN feature, but some material attributes like wool and feather gave the best result with LBP feature as shown in figures 4.19. and 4.20.. The object parts like rein, saddle, window, row wind, wheel, handlebars, engine gave the best results with CNN feature as shown in figures 4.18. and 4.19.. The spatial attributes like

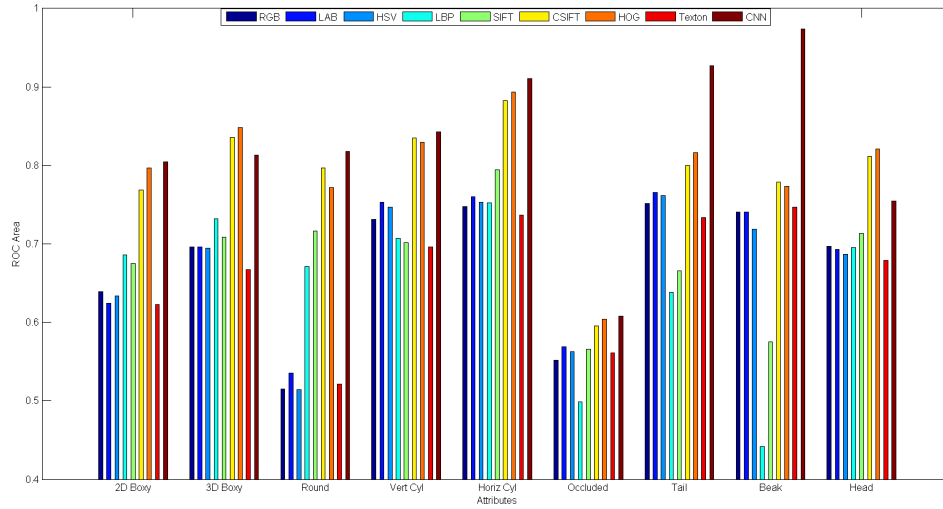


FIGURE 4.9.: The results of attributes and low-level features and mid-level features correlations for a-Pascal dataset, Part-1

2D boxy, 3D boxy, vertical cylinder, horizontal cylinder gave changeable results with HOG, CSIFT and CNN features as shown in figure 4.16..

In Shoes dataset CNN feature is the best feature to learning all attributes as shown in figure 4.21.. For this dataset HOG feature also gave good results for all attributes. The CSIFT feature is the third-best feature in this dataset, but SIFT feature worked better in 'log-on-the-leg' attribute than CSIFT descriptors. The LBP and SIFT features performed on 'covered-with-ornaments' is lower than other attributes. The color histograms gave the worst results on 'open' attribute.

In Attributes of People dataset also CNN feature provide the best results all of them except has-glasses attribute which worked better with LBP feature as shown in figure 4.22.. The CSIFT descriptor may be the second-best feature for this dataset except for 'has-glasses' and 'has-shorts' attributes. The HOG feature is the third-best feature for Attributes of People dataset. The color histograms gave the worst results on 'has-glasses' attribute.

## Chapter 4. EXPERIMENTS & RESULTS

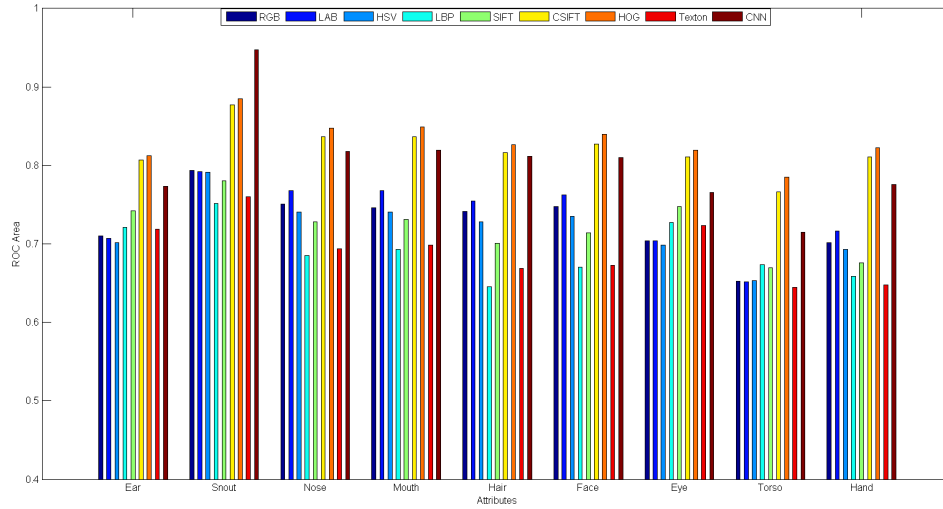


FIGURE 4.10.: The results of attributes and low-level features and mid-level features correlations for a-Pascal dataset, Part-2

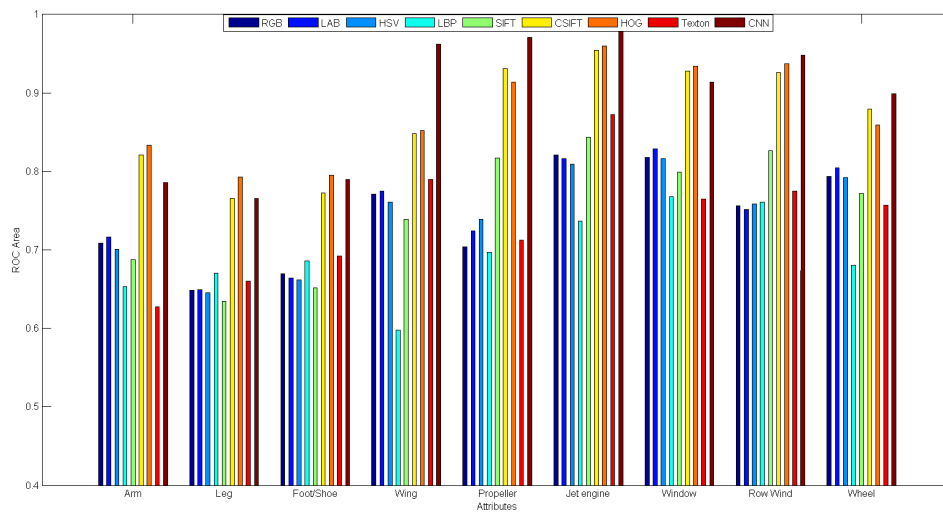


FIGURE 4.11.: The results of attributes and low-level features and mid-level features correlations for a-Pascal dataset, Part-3

Chapter 4. *EXPERIMENTS & RESULTS*

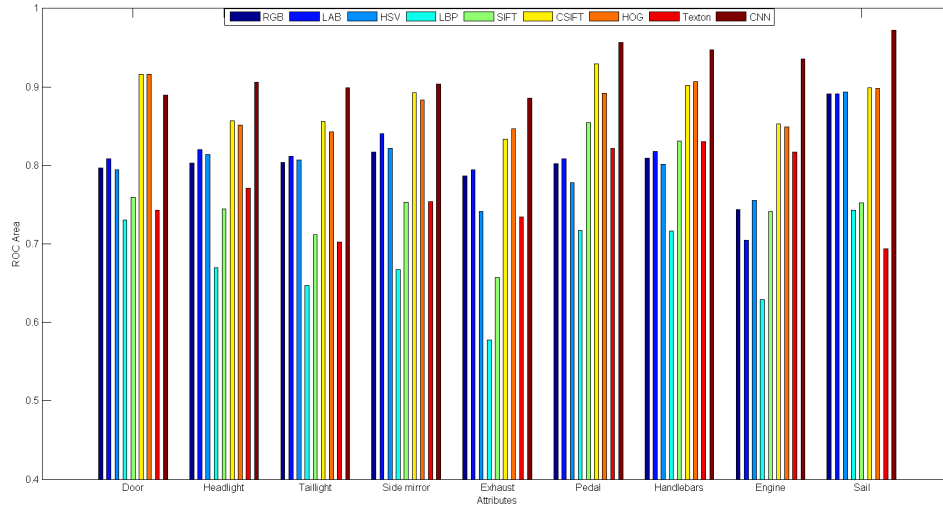


FIGURE 4.12.: The results of attributes and low-level features and mid-level features correlations for a-Pascal dataset, Part-4

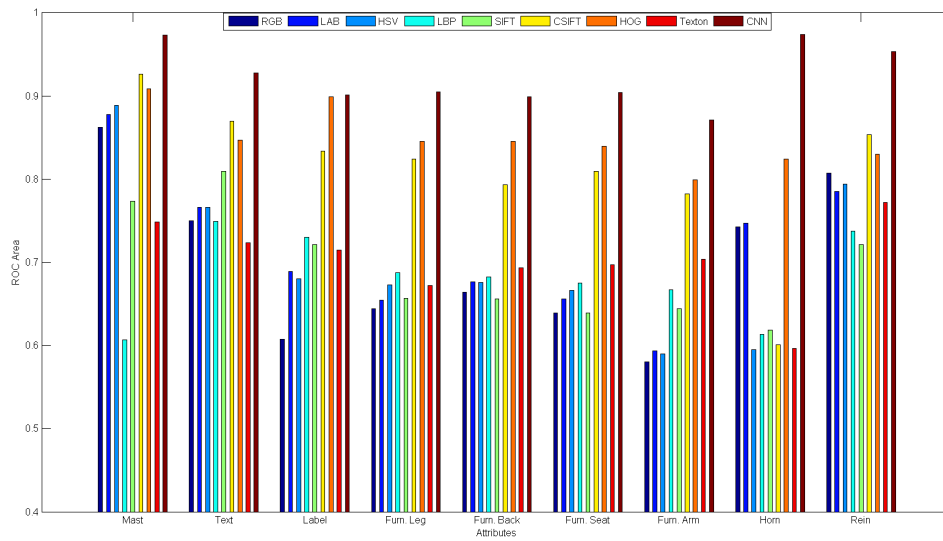


FIGURE 4.13.: The results of attributes and low-level features and mid-level features correlations for a-Pascal dataset, Part-5



## Chapter 4. EXPERIMENTS & RESULTS

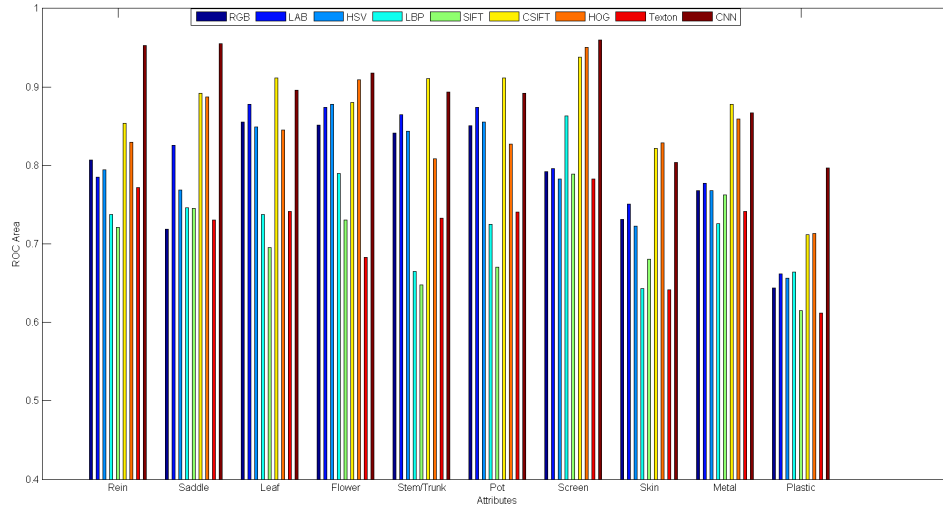


FIGURE 4.14.: The results of attributes and low-level features and mid-level features correlations for a-Pascal dataset, Part-6

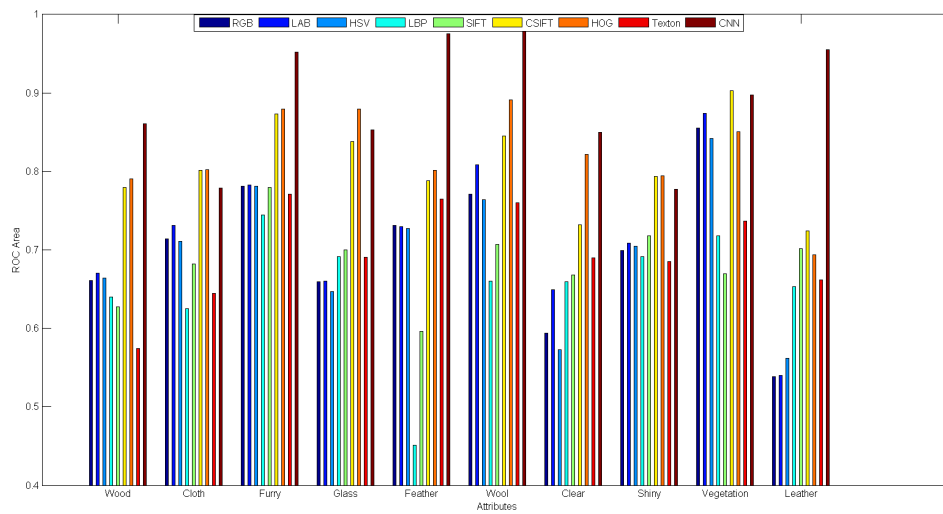


FIGURE 4.15.: The results of attributes and low-level features and mid-level features correlations for a-Pascal dataset, Part-7

Chapter 4. *EXPERIMENTS & RESULTS*

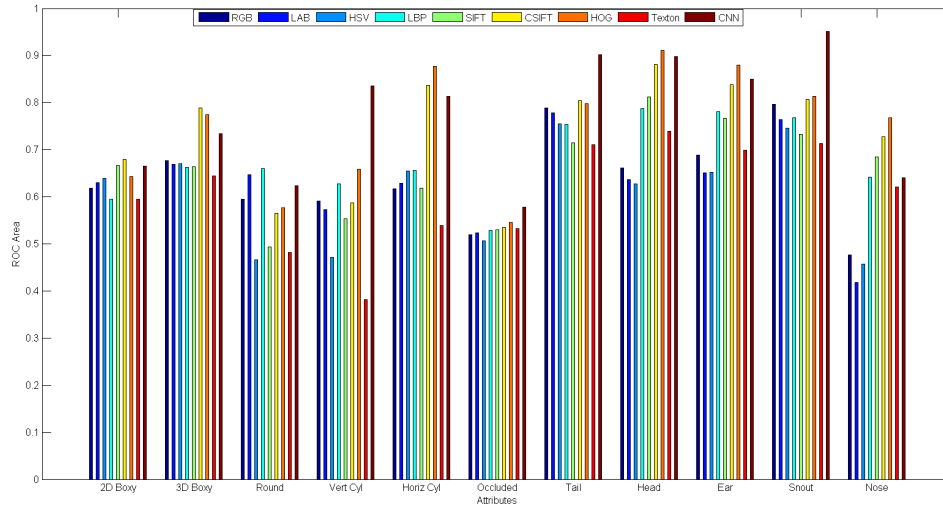


FIGURE 4.16.: The results of attributes and low-level features and mid-level features correlations for a-Yahoo dataset, Part-1

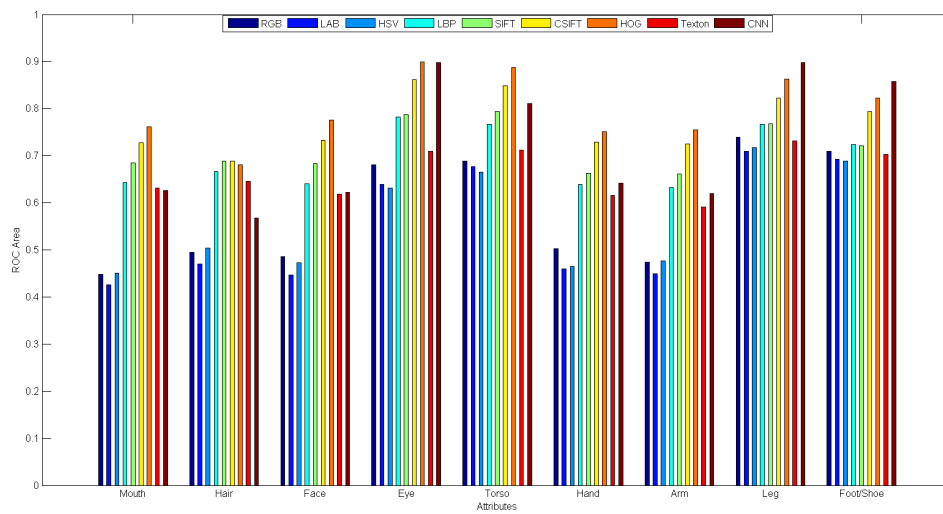


FIGURE 4.17.: The results of attributes and low-level features and mid-level features correlations for a-Yahoo dataset, Part-2

Chapter 4. *EXPERIMENTS & RESULTS*

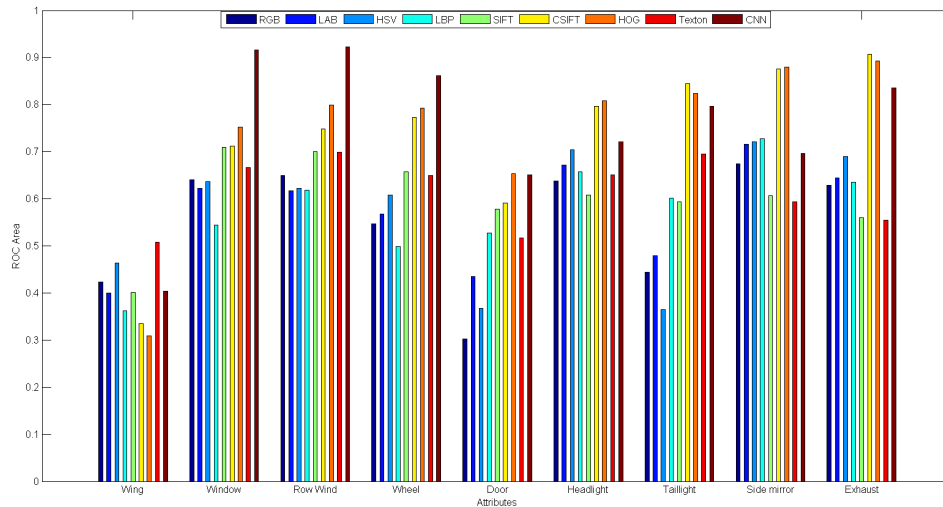


FIGURE 4.18.: The results of attributes and low-level features and mid-level features correlations for a-Yahoo dataset, Part-3

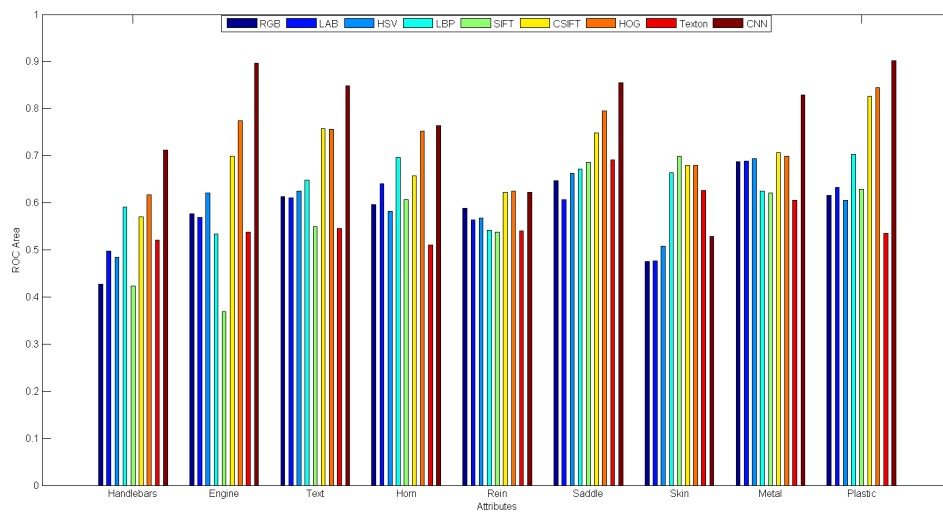


FIGURE 4.19.: The results of attributes and low-level features and mid-level features correlations for a-Yahoo dataset, Part-4

## Chapter 4. EXPERIMENTS & RESULTS

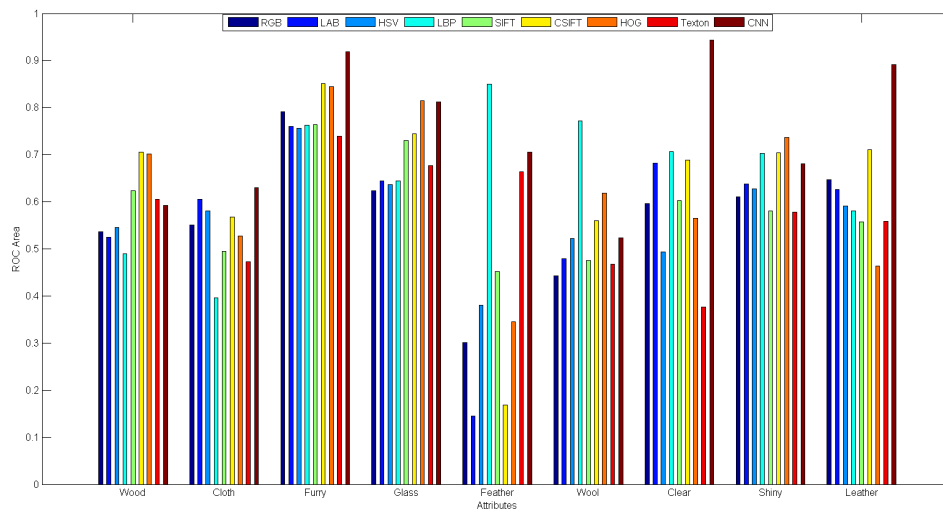


FIGURE 4.20.: The results of attributes and low-level features and mid-level features correlations for a-Yahoo dataset, Part-5

Chapter 4. *EXPERIMENTS & RESULTS*

pointy-at-the-front	paf
open	op
bright-in-color	bc
covered-with-ornaments	co
shiny	sh
high-at-the-heel	htl
long-on-the-leg	lol
formal	fo
sporty	sp
feminine	fe

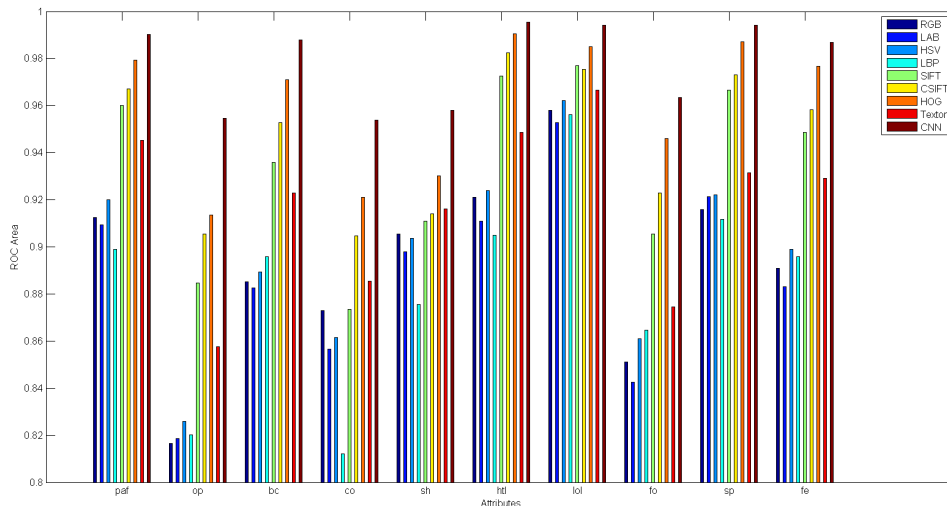


FIGURE 4.21.: The results of attributes and low-level features and mid-level features correlations for Shoes dataset

Chapter 4. *EXPERIMENTS & RESULTS*

is-male	im
has-long-hair	hlh
has-glasses	hg
has-hat	hh
has-t-shirt	hts
has-long-sleeves	hls
has-shorts	hs
has-jeans	hj
has-long-pants	hlp

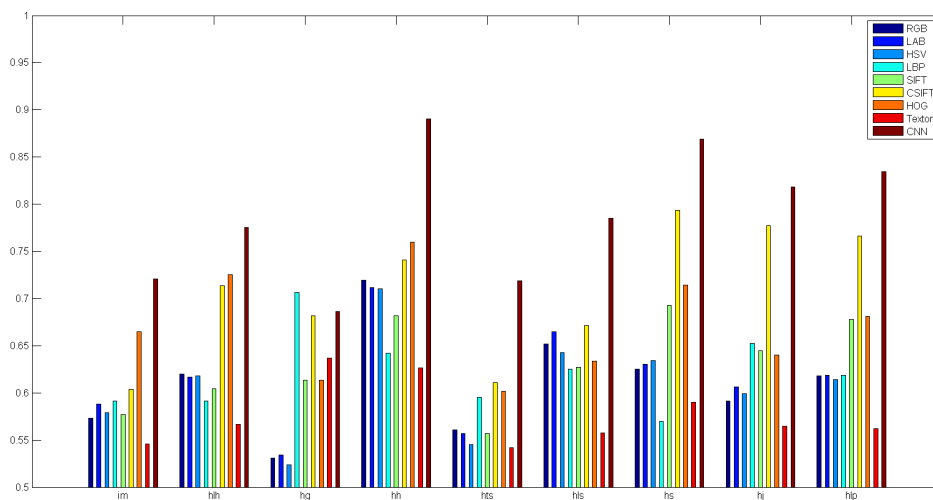


FIGURE 4.22.: The results of attributes and low-level features and mid-level features correlations for Attributes of People dataset

### **Performance of Feature Combinations**

- **Results on a-Pascal Dataset** In figure 4.5. there are ROC curves results for each feature. After the evaluation of performance of individual features, we look at some combinations of these features. In forming these combinations, we employed both early fusion and late fusion techniques. In these evaluations, we used the best color result that LAB color for a-Pascal dataset, CSIFT, HOG, LBP, Texton and CNN features. We did not use SIFT features because CSIFT features gave better results than it and their structure is similar. The results for all of these combinations are in Tables 4.15, 4.14 and ROC curves are in Figure 4.24.. Note that combinations of features were used firstly in a-Pascal dataset and best result of these combinations was applied to other datasets.

The following conclusions can be done through by these results. Only using the shape features or only using the color features are generally inadequate to represent images. We added to CSIFT and HOG combinations texton, LBP and LAB color. The LAB color feature made the best contribution. Color histograms are insufficient to represent an image, but they are very effective to complete shape features. We can say using more features do not always give the best result. The features combination except CNN is not the best result. For this dataset combination of CSIFT + HOG + LAB Color gave the best result. We added the CNN feature to this combination and the performance of this combination decreased. We wanted to see better result for these combinations, so we used late fusion to see combination clearly and this combination gave the best result for this dataset as shown in Table 4.15. As it can be seen in these results, Weighted Late Fusion (WLF) gives the best results in obtaining the combinations of features.

- **Results on a-Yahoo Dataset** We used the best combination of a-Pascal dataset which CSIFT + HOG + Color for a-Yahoo dataset. But differently we used RGB Color which the best color result for this dataset as shown in 4.2. Then, we used CNN features in combinations with early and late fusion methods. The table 4.16 and the figure 4.28. shows the results. Only using CNN feature gave better result than early fusion of CNN.

Chapter 4. *EXPERIMENTS & RESULTS*

Features	ROC Area	AP
CNN + CSIFT* + HOG*	0.858	0.537
CNN + CSIFT + HOG + LAB Color	0.864	0.539
CNN + CSIFT*	0.869	0.523
CNN + CSIFT* + HOG* + LAB Color	0.865	0.547
CSIFT + Texton	0.881	0.510
CSIFT + HOG + Texton	0.882	0.546
HOG + LAB Color + Texton	0.886	0.534
LBP + HOG + LAB Color	0.884	0.531
CSIFT + LBP + HOG	0.893	0.557
CSIFT + LAB Color + HOG + Texton	0.897	0.566
CSIFT* + HOG* + LAB Color	0.898	0.573
CSIFT + LBP + HOG + Texton	0.899	0.564
CSIFT + HOG + LAB Color + Texton + LBP	0.901	0.570
CSIFT + HOG + LAB Color	<b>0.904</b>	0.574

TABLE 4.14: The results of early fusion feature combinations for a-Pascal dataset  
\* : 750 kmeans centers

Features	Fusion Type	ROC Area	AP
CNN + LAB Color	LF	0.891	0.522
	WLF	0.892	0.521
CNN + CSIFT*	LF	0.915	0.593
	WLF	0.914	0.589
CNN + HOG*	LF	0.916	0.602
	WLF	0.915	0.599
CNN + CSIFT* + LAB Color	LF	0.916	0.596
	WLF	0.916	0.597
CNN + HOG* + LAB Color	LF	0.919	0.609
	WLF	0.920	0.609
CNN + (CSIFT* + HOG* + LAB Color)	LF	0.923	0.621
CNN + HOG* + CSIFT*	LF	0.924	0.624
	WLF	0.924	0.624
CNN + (CSIFT + HOG + LAB Color)	LF	0.925	0.620
CNN + CSIFT* + HOG* + LAB Color	LF	0.925	0.625
	WLF	<b>0.926</b>	0.627

TABLE 4.15: The results of late fusion and weighted late fusion feature combinations for a-Pascal dataset  
\* : 750 kmeans centers  
LF : Late Fusion, WLF : Weighted Late Fusion

- **Results on Shoes Dataset** The best combination of a-Pascal dataset was used for this dataset too. We used HSV color for these combinations because it gave the best result for this dataset as shown in 4.3. The results are in the table 4.17 and in the figure 4.29..



Chapter 4. *EXPERIMENTS & RESULTS*

Features	Fusion Types	ROC Area	AP
CNN + (CSIFT + HOG + RGB Color)	EF	0.785	0.436
CNN + (CSIFT* + HOG* + LAB Color)	EF	0.795	0.461
CNN + LAB Color	LF	0.806	0.440
CNN + RGB Color	LF	0.810	0.444
	WLF	0.814	0.453
CSIFT* + HOG* + LAB Color	EF	0.824	0.439
CSIFT + HOG + RGB Color	EF	0.832	0.443
CNN + CSIFT + RGB Color	LF	0.847	0.514
	WLF	0.850	0.522
CNN + CSIFT	LF	0.858	0.537
	WLF	0.858	0.536
CNN + HOG* + RGB Color	LF	0.861	0.546
	WLF	0.863	0.552
CNN + (CSIFT* + HOG* + LAB Color)	LF	0.863	0.553
CNN + (CSIFT + HOG + RGB Color)	LF	0.866	0.556
CNN + HOG*	LF	0.870	0.561
	WLF	0.869	0.559
CNN + HOG* + CSIFT + RGB Color	LF	0.868	0.546
	WLF	0.870	0.553
CNN + HOG* + CSIFT	LF	0.875	0.559
	WLF	<b>0.875</b>	0.561

TABLE 4.16: The results of feature combinations for a-Yahoo dataset  
\* 750 kmeans centers

EF : Early Fusion, LF : Late Fusion, WLF : Weighted Late Fusion

Features	Fusion Types	ROC Area	AP
CNN + HSV Color	LF	0.978	0.974
	WLF	0.979	0.975
CNN + HSV Color + CSIFT	LF	0.979	0.974
	WLF	0.979	0.974
CSIFT + HOG + HSV Color	EF	0.979	0.976
CNN + CSIFT + HOG + HSV Color	LF	0.980	0.976
	WLF	0.980	0.977
CNN + CSIFT	LF	0.981	0.976
	WLF	0.981	0.976
CNN + HOG + CSIFT	LF	0.981	0.978
	WLF	0.982	0.978
CNN + HOG + HSV Color	LF	0.981	0.978
	WLF	0.982	0.978
CNN + HOG	LF	0.983	0.980
	WLF	0.983	0.980
CNN + (CSIFT + HOG + HSV Color)	EF	0.984	0.981
CNN + (CSIFT + HOG + HSV Color)	LF	<b>0.987</b>	0.985

TABLE 4.17: The results of feature combinations for Shoes dataset  
EF : Early Fusion, LF : Late Fusion, WLF : Weighted Late Fusion

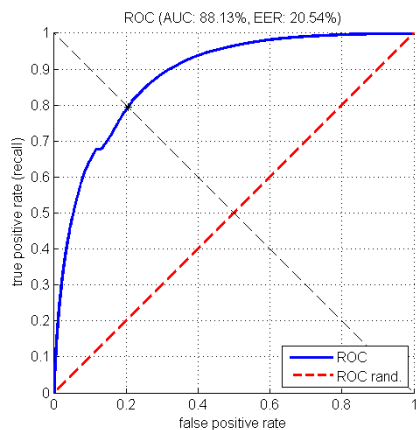
- **Results on Attribute of People Dataset** We used the best combination of a-Pascal dataset with LAB color for this dataset. Early and late fusion methods were used for CNN feature. The results are in the table 4.18 and in the figure 4.30.. The early fusion of CNN performance was worse than only using of CNN feature.

Features	Fusion Types	ROC Area	AP
CSIFT + HOG + LAB Color	EF	0.687	0.384
CNN + (CSIFT + HOG + LAB Color)	EF	0.783	0.474
CNN + LAB Color	LF	0.800	0.494
	WLF	0.803	0.496
CNN + HOG + LAB Color	LF	0.805	0.506
	WLF	0.809	0.509
CNN + HOG	LF	0.809	0.509
	WLF	0.811	0.512
CNN + LAB Color + CSIFT	LF	0.809	0.510
	WLF	0.812	0.513
CNN + HOG + CSIFT	LF	0.813	0.518
	WLF	0.816	0.521
CNN + CSIFT	LF	0.816	0.518
	WLF	0.817	0.520
CNN + (CSIFT + HOG + LAB Color)	LF	0.809	0.520
CNN + CSIFT + HOG + LAB Color	LF	0.809	0.512
	WLF	<b>0.812</b>	0.516

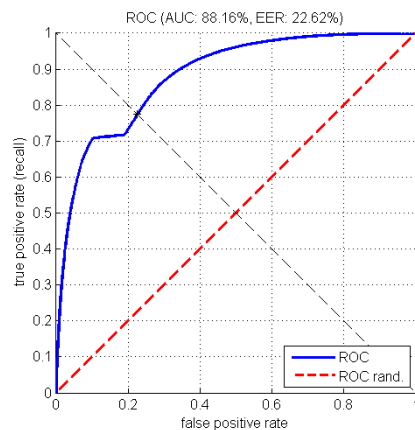
TABLE 4.18: The results of feature combinations for Attributes of People dataset  
EF : Early Fusion, LF : Late Fusion, WLF : Weighted Late Fusion

Except for the aYahoo dataset, the best results are obtained using the (weighted) late fusion of CSIFT, HOG, Color and CNN features. We can say that while CNN features are powerful supervised features, they may not achieve the best results, and the combination of low-level features with CNN features gives promising improvements on attribute recognition

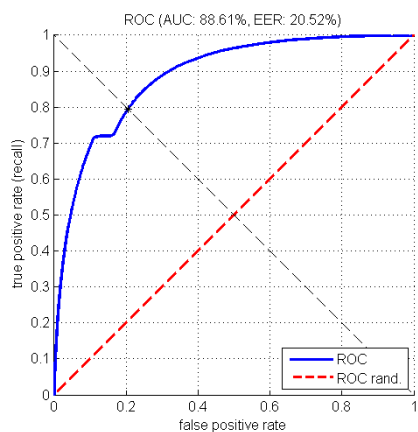
Chapter 4. *EXPERIMENTS & RESULTS*



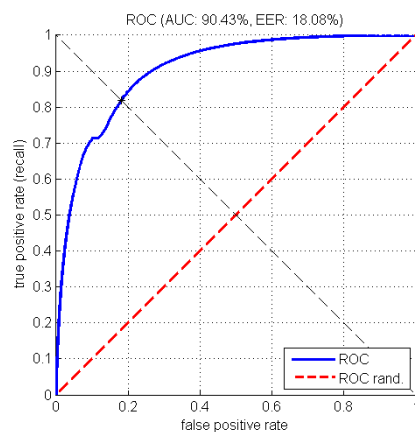
(A) CSIFT + Texton for a-Pascal dataset



(B) CSIFT + HOG + Texton for a-Pascal dataset



(C) Hog + Texton + LAB Color for a-Pascal dataset



(D) CSIFT + HOG + LAB Color for a-Pascal dataset

FIGURE 4.23.: ROC curves of feature combinations for a-Pascal dataset - Part 1

Chapter 4. *EXPERIMENTS & RESULTS*

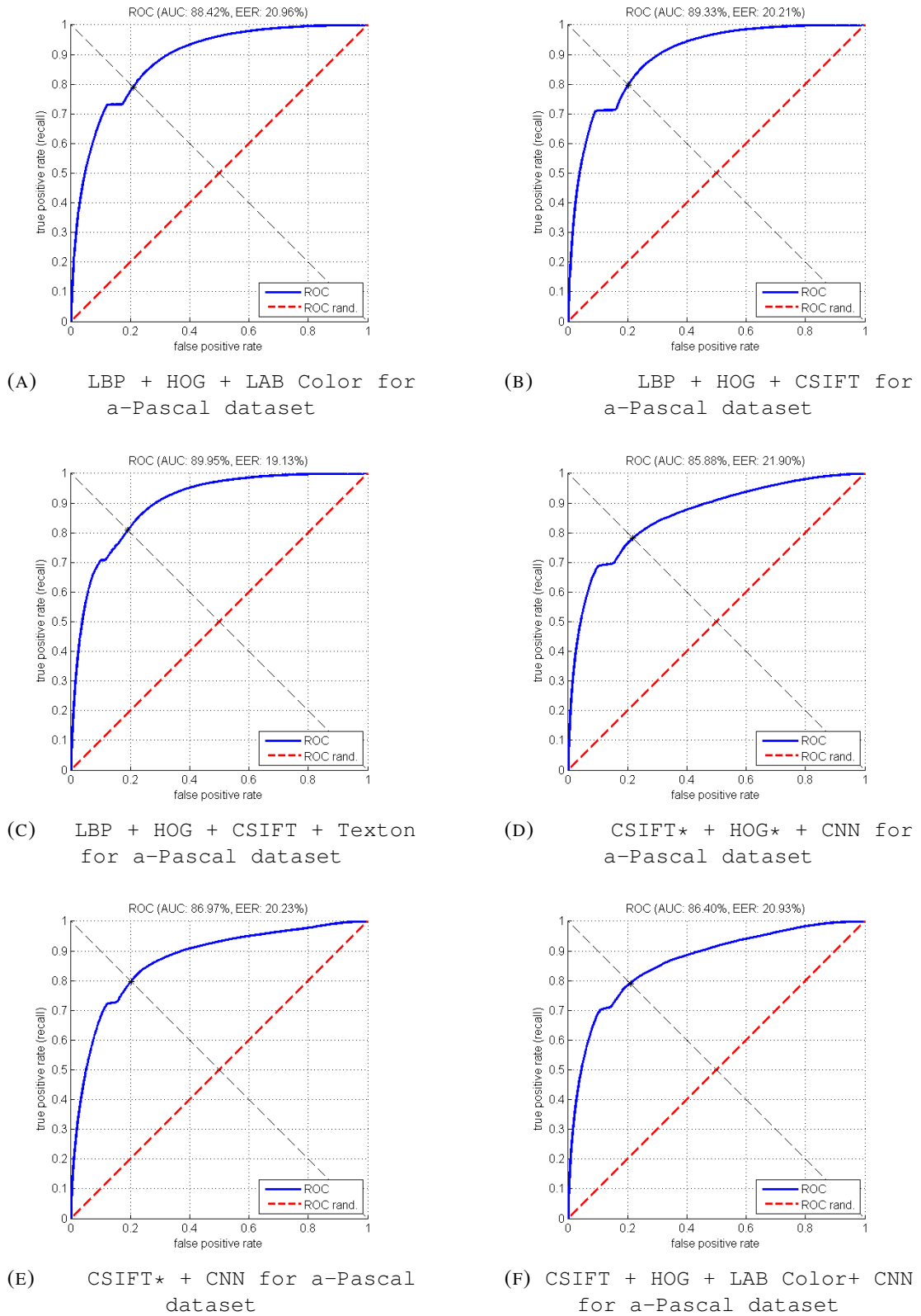
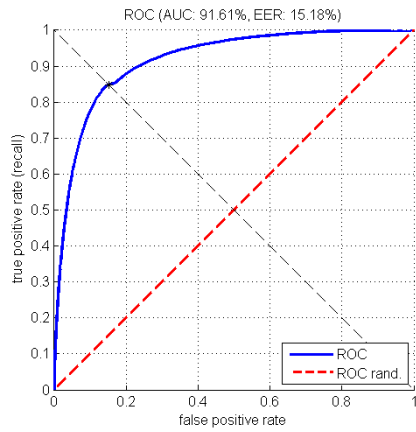
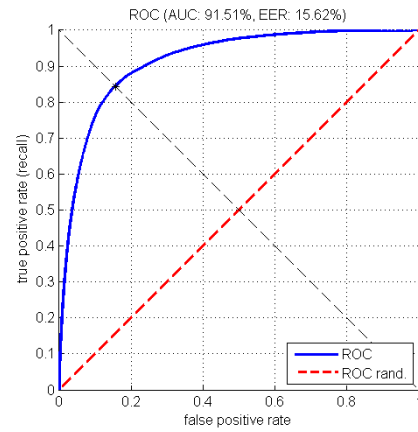


FIGURE 4.24.: ROC curves of feature combinations for a-Pascal dataset - Part 2

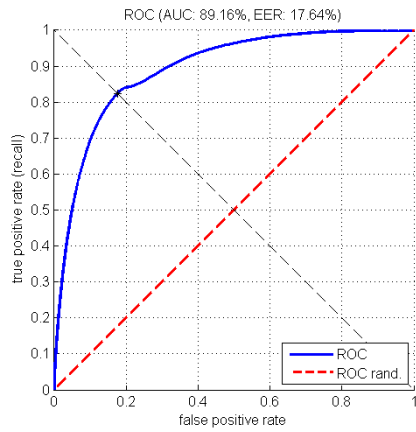
Chapter 4. *EXPERIMENTS & RESULTS*



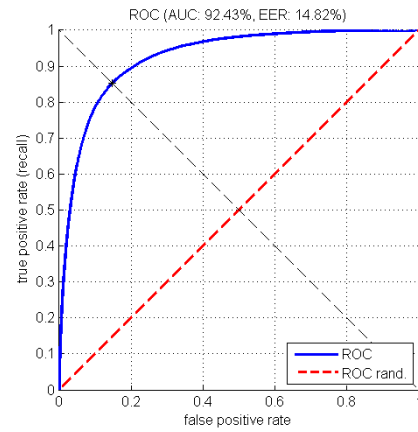
(A) CNN + HOG\* (LF) for a-Pascal dataset



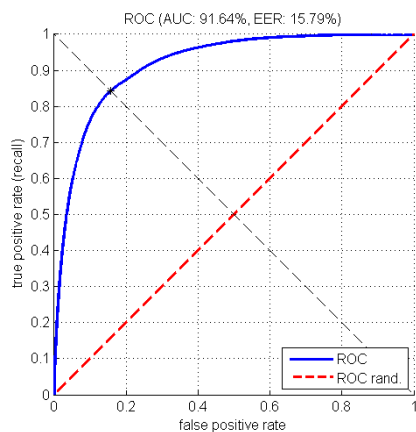
(B) CNN + CSIFT\* (LF) for a-Pascal dataset



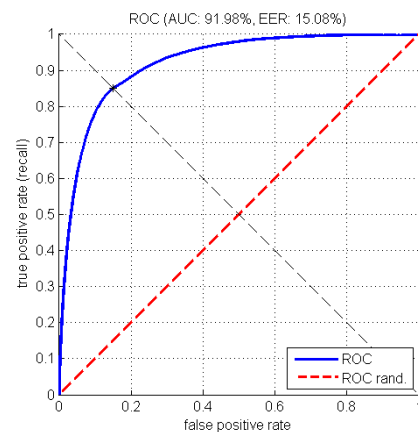
(C) CNN + LAB Color (LF) for a-Pascal dataset



(D) CNN + CSIFT\* + HOG\* (LF) for a-Pascal dataset

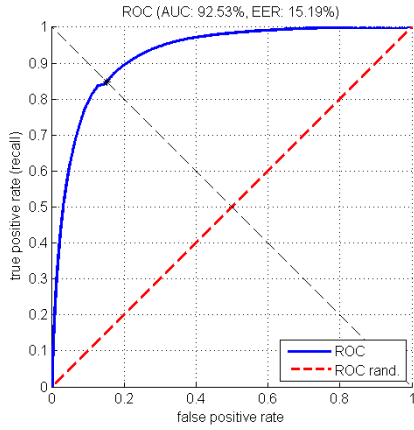


(E) CNN + CSIFT\* + LAB Color (LF) for a-Pascal dataset

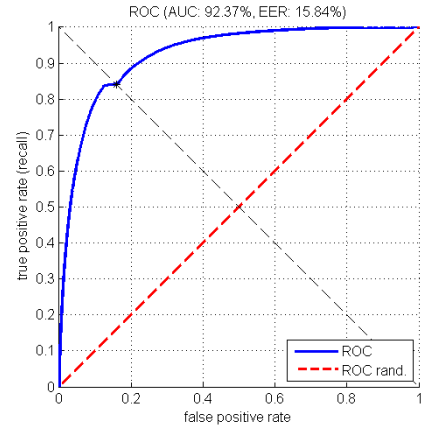


(F) CNN + HOG\* + LAB Color (LF) for a-Pascal dataset

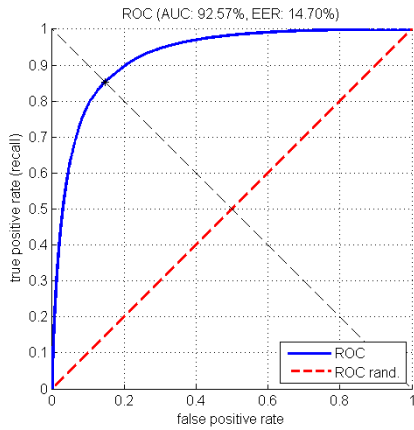
FIGURE 4.25.: ROC curves of late fusion feature combinations for a-Pascal dataset - Part I



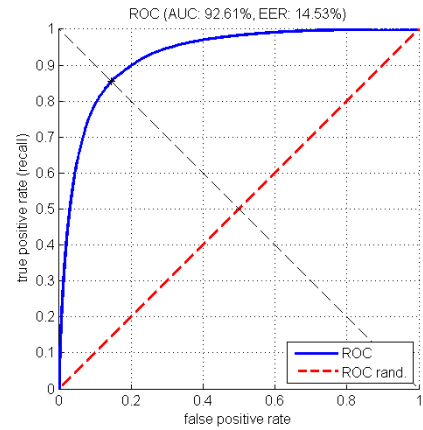
(A) (CSIFT + HOG + LAB Color) + CNN (LF) for a-Pascal dataset



(B) (CSIFT + HOG + LAB Color) + CNN (LF) for a-Pascal dataset



(C) CSIFT\* + HOG\* + LAB Color + CNN (LF) for a-Pascal dataset



(D) CSIFT\* + HOG\* + LAB Color + CNN (WLF) for a-Pascal dataset

FIGURE 4.26.: ROC curves of late fusion feature combinations for a-Pascal dataset - Part2

Chapter 4. *EXPERIMENTS & RESULTS*

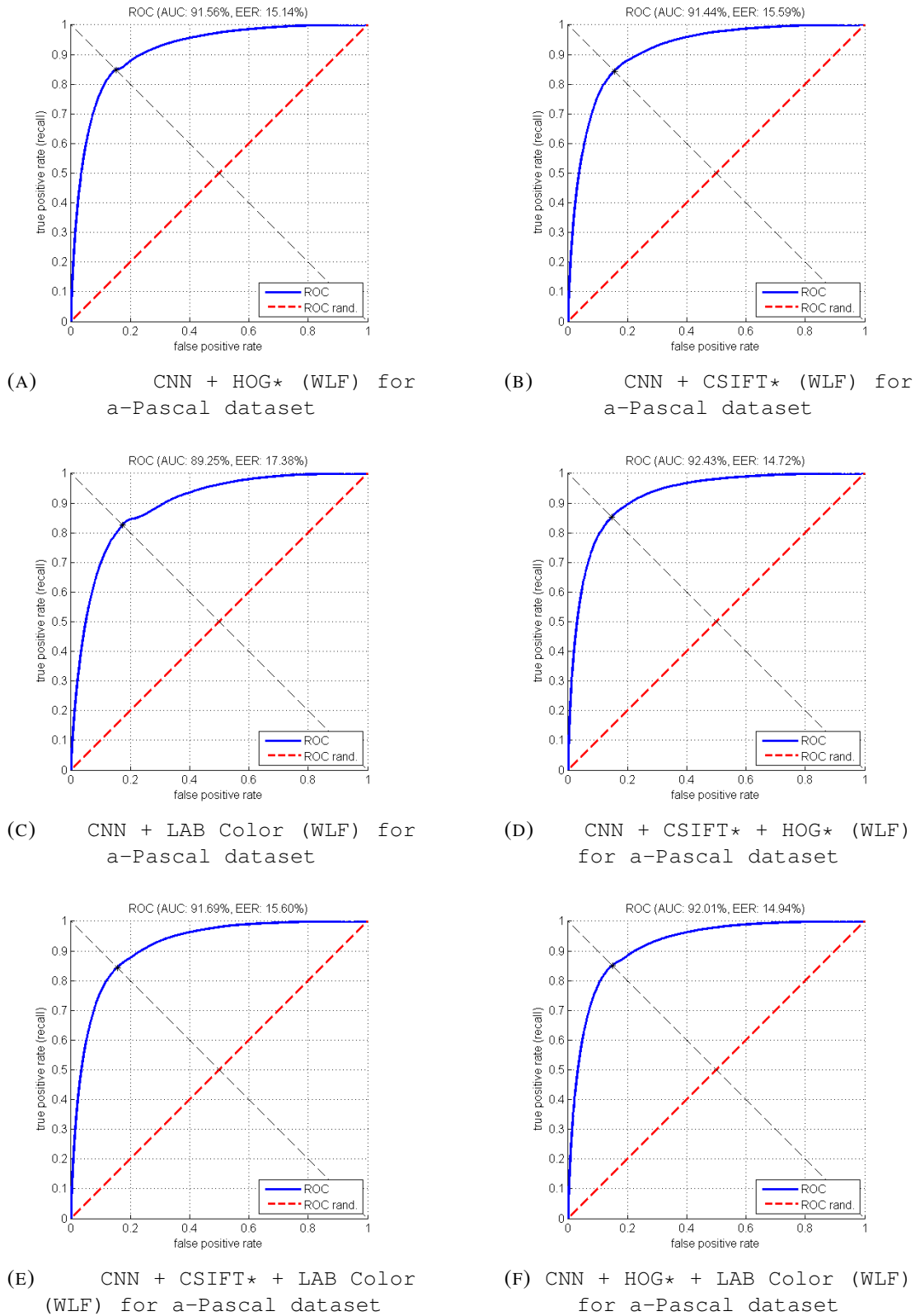
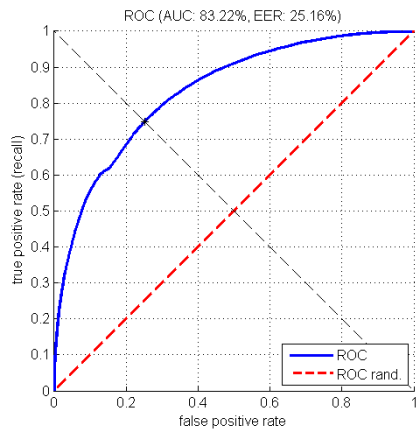
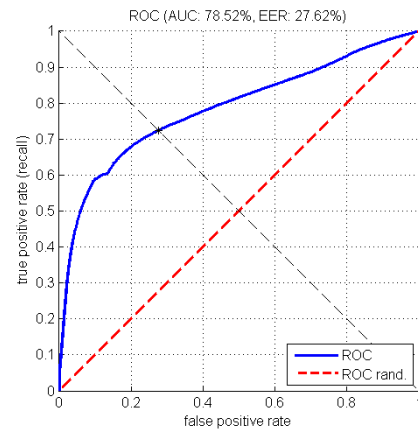


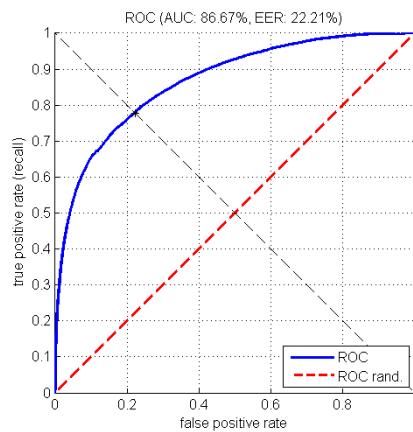
FIGURE 4.27.: ROC curves of wighted late fusion feature combinations for a-Pascal dataset



(A) CSIFT + HOG + RGB Color for a-Yahoo dataset



(B) CSIFT + HOG + RGB Color+ CNN for a-Yahoo dataset

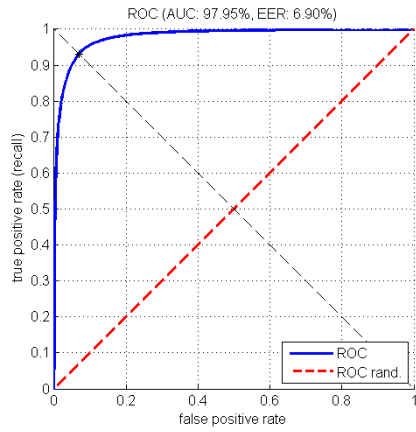


(C) CSIFT + HOG + RGB Color+ CNN (Late Fusion) for a-Yahoo dataset

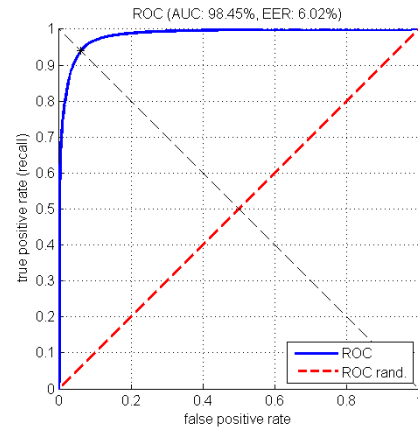
FIGURE 4.28.: ROC curves of feature combinations for a-Yahoo dataset



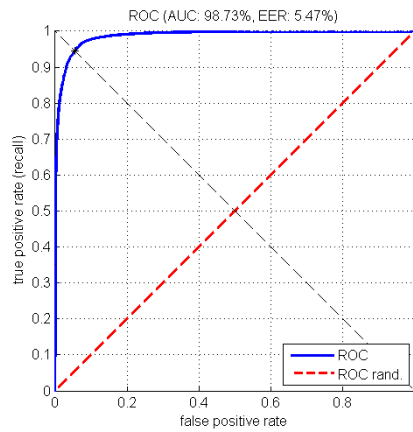
Chapter 4. *EXPERIMENTS & RESULTS*



(A) CSIFT + HOG + HSV Color for Shoes dataset

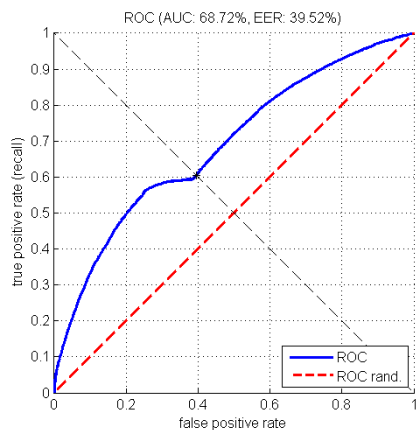


(B) CSIFT + HOG + HSV Color+ CNN for Shoes dataset

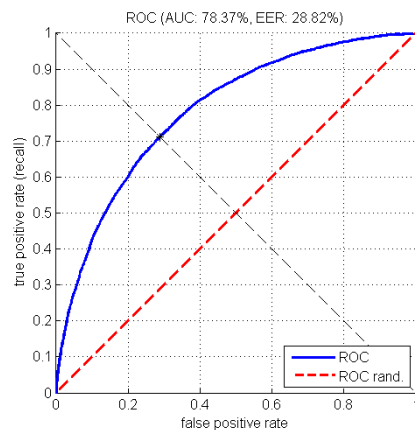


(C) CSIFT + HOG + HSV Color+ CNN (Late Fusion) for Shoes dataset

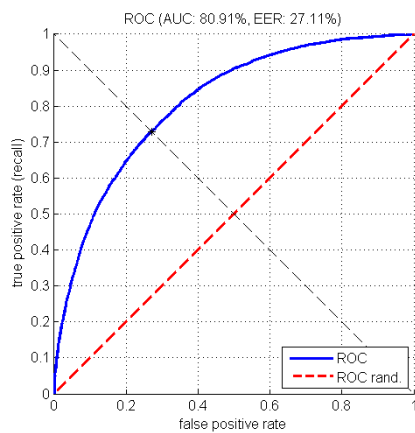
FIGURE 4.29.: ROC curves of feature combinations for Shoes dataset



(A) CSIFT + HOG + RGB Color for Attributes of People dataset



(B) CSIFT + HOG + RGB Color + CNN for Attributes of People dataset



(C) CSIFT + HOG + RGB Color + CNN (Late Fusion) for Attributes of People dataset

FIGURE 4.30.: ROC curves of feature combinations for People dataset

#### 4.4. Comparison with Reference Work

We used the Farhadi et al. [1] method in this work. Their results for logistic regression and selected features were compared with our results in table 4.19. In their work they combined only three features that HOG, LAB Color and Texton. We used different combinations of them and with other features. As seen from the table the results can be improved with different features and their combinations.

<b>Features</b>	<b>AP</b>
HOG + LAB Color + Texton [1]	0.535
HOG + LAB Color + Texton	0.534
CSIFT + HOG + LAB Color	0.574
CNN + CSIFT* + HOG* + LAB Color (LF)	0.625
CNN + CSIFT* + HOG* + LAB Color (WLF)	0.627

TABLE 4.19: The results of comparison between [1] and our work for a-Pascal dataset

Lampert et al. [10] used the a-Pascal dataset too. The table 4.20 shows the comparison the results of two methods. They used similar features like color SIFT as rgSIFT color histograms PHOG, but the results are lower. Different methods could cause this result. They used direct attribute prediction (DAP) and indirect attribute prediction (IAP) differently from us. Another reason can be the feature combinations. More features can decrease the results as seen in the results of this work.

<b>Features</b>	<b>ROC Area</b>
HSV Color + SIFT + rgSIFT + PHOG + SURF [10]	0.737
CSIFT + HOG + LAB Color	0.904
CNN + CSIFT* + HOG* + LAB Color (LF)	0.925
CNN + CSIFT* + HOG* + LAB Color (WLF)	0.926

TABLE 4.20: The results of comparison between [10] and our work for a-Pascal dataset

## Chapter 4. *EXPERIMENTS & RESULTS*

Bourdev et al. [3] used HSV color histograms, HOG and skin-specific features to describe people by attributes. They wanted to learn invariant attributes to viewpoints and poses. Our method does not provide these properties. Our result is lower than their result as shown in 4.21.

Razavian et al. [27] also used CNN features in their work. The table 4.21 shows their results are better than us. They extracted CNN feature vector within the bounding box bins we, on the other hand used full size of image when extracting CNN feature vectors and this could reduce the performance.

<b>Features</b>	<b>AP</b>
CNN (OverFeat) [27]	0.730
HSV Color + HOG + Skin-specific features [3]	0.651
CNN + (CSIFT + HOG + LAB Color)	0.474
CNN + CSIFT + HOG + LAB Color (LF)	0.512
CNN + CSIFT + HOG + LAB Color (WLF)	0.516

TABLE 4.21: The results of comparison between [27], [3] and our work for Attributes of People dataset

## 5. CONCLUSIONS

In this thesis, our aim is to analyze the effects of low-level features for visual attribute recognition. To do this, we explore four main categories of low-level features, which are RGB, HSV and LAB color histograms for color features, texton and LBP for texture features, HOG and SIFT for shape features, CSIFT for hybrid and CNN for deep learning features.

In our experiments, we make use of four datasets, which are designed to handle different visual recognition tasks, such as object recognition, shoe description and people description. These are a-Pascal, a-Yahoo, Shoes and Attributes of People datasets.

Overall the CNN feature can be the most effective feature for attribute learning. CSIFT and HOG features are the second best features and may be replaced with each other. The structure of attributes affects the results of learning. For example, the body parts worked better with HOG feature and human clothes worked better with CNN feature.

We also look at the performance of some feature combinations. In general, using features in combination gives better performance. However, the results also show us that using more features for combinations does not give better results. It is important to find feature combinations such that features complement each other.

Based on these results CNN feature could be used in product search which can be shoes or clothes. HOG, CSIFT or CNN may be preferred for object recognition problems.

This work can be extended by different datasets and different features. The feature combinations can be handled more detailed. We used the best combinations of a-Pascal dataset for other datasets. All combinations of features can be applied to each dataset separately and combination analyzing can be done according to datasets.

## REFERENCES

- [1] Ali Farhadi, Ian Endres, Derek Hoiem, and David A. Forsyth. Describing objects by their attributes. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 1778–1785, 2009. doi: 10.1109/CVPRW.2009.5206772. URL <http://dx.doi.org/10.1109/CVPRW.2009.5206772>.
- [2] Adriana Kovashka, Devi Parikh, and Kristen Grauman. Whittlesearch: Image search with relative attribute feedback. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 2973–2980, 2012. doi: 10.1109/CVPR.2012.6248026. URL <http://dx.doi.org/10.1109/CVPR.2012.6248026>.
- [3] Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. Describing people: A poselet-based approach to attribute classification. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 1543–1550, 2011. doi: 10.1109/ICCV.2011.6126413. URL <http://dx.doi.org/10.1109/ICCV.2011.6126413>.
- [4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 1106–1114, 2012. URL <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-discretionary{-}{}{}-neural-networks>.
- [5] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 951–958, 2009. doi: 10.1109/CVPRW.2009.5206594. URL <http://dx.doi.org/10.1109/CVPRW.2009.5206594>.

- [6] Neeraj Kumar, Peter N. Belhumeur, and Shree K. Nayar. Facetracer: A search engine for large collections of images with faces. In *Computer Vision - ECCV 2008, 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV*, pages 340–353, 2008. doi: 10.1007/978-3-540-88693-8\_25. URL [http://dx.doi.org/10.1007/978-3-540-88693-8\\_25](http://dx.doi.org/10.1007/978-3-540-88693-8_25).
- [7] Devi Parikh and Kristen Grauman. Relative attributes. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 503–510, 2011. doi: 10.1109/ICCV.2011.6126281. URL <http://dx.doi.org/10.1109/ICCV.2011.6126281>.
- [8] Amir Sadvnik, Andrew C. Gallagher, Devi Parikh, and Tsuhan Chen. Spoken attributes: Mixing binary and relative attributes to say the right thing. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 2160–2167, 2013. doi: 10.1109/ICCV.2013.268. URL <http://dx.doi.org/10.1109/ICCV.2013.268>.
- [9] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 365–372, 2009. doi: 10.1109/ICCV.2009.5459250. URL <http://dx.doi.org/10.1109/ICCV.2009.5459250>.
- [10] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(3):453–465, 2014. doi: 10.1109/TPAMI.2013.140. URL <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2013.140>.
- [11] Dinesh Jayaraman and Kristen Grauman. Zero-shot recognition with unreliable attributes. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3464–3472, 2014. URL <http://papers.nips.cc/paper/5290-zero-shot-recognition-with-unreliable-attributes>.

- [12] Vittorio Ferrari and Andrew Zisserman. Learning visual attributes. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 433–440, 2007. URL <http://papers.nips.cc/paper/3217-learning-visual-attributes>.
- [13] Mohammad Rastegari, Ali Farhadi, and David A. Forsyth. Attribute discovery via predictable discriminative binary codes. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI*, pages 876–889, 2012. doi: 10.1007/978-3-642-33783-3\_63. URL [http://dx.doi.org/10.1007/978-3-642-33783-3\\_63](http://dx.doi.org/10.1007/978-3-642-33783-3_63).
- [14] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 133–142, 2002. doi: 10.1145/775047.775067. URL <http://doi.acm.org/10.1145/775047.775067>.
- [15] Daniel A. Vaquero, Rogério Schmidt Feris, Duan Tran, Lisa M. G. Brown, Arun Hampapur, and Matthew Turk. Attribute-based people search in surveillance environments. In *IEEE Workshop on Applications of Computer Vision (WACV 2009), 7-8 December, 2009, Snowbird, UT, USA*, pages 1–8, 2009. doi: 10.1109/WACV.2009.5403131. URL <http://dx.doi.org/10.1109/WACV.2009.5403131>.
- [16] Rogerio Feris, Russell Bobbitt, Lisa M. G. Brown, and Sharath Pankanti. Attribute-based people search: Lessons learnt from a practical surveillance system. In *International Conference on Multimedia Retrieval, ICMR '14, Glasgow, United Kingdom - April 01 - 04, 2014*, page 153, 2014. doi: 10.1145/2578726.2578732. URL <http://doi.acm.org/10.1145/2578726.2578732>.
- [17] Amar Parkash and Devi Parikh. Attributes for classifier feedback. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III*, pages 354–368, 2012. doi: 10.1007/978-3-642-33712-3\_26. URL [http://dx.doi.org/10.1007/978-3-642-33712-3\\_26](http://dx.doi.org/10.1007/978-3-642-33712-3_26).



- [18] Adriana Kovashka, Sudheendra Vijayanarasimhan, and Kristen Grauman. Actively selecting annotations among objects and attributes. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 1403–1410, 2011. doi: 10.1109/ICCV.2011.6126395. URL <http://dx.doi.org/10.1109/ICCV.2011.6126395>.
- [19] Arijit Biswas and Devi Parikh. Simultaneous active learning of classifiers & attributes via relative feedback. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 644–651, 2013. doi: 10.1109/CVPR.2013.89. URL <http://dx.doi.org/10.1109/CVPR.2013.89>.
- [20] Lucy Liang and Kristen Grauman. Beyond comparing image pairs: Setwise active learning for relative attributes. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 208–215, 2014. doi: 10.1109/CVPR.2014.34. URL <http://dx.doi.org/10.1109/CVPR.2014.34>.
- [21] Walter J. Scheirer, Neeraj Kumar, Peter N. Belhumeur, and Terrance E. Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 2933–2940, 2012. doi: 10.1109/CVPR.2012.6248021. URL <http://dx.doi.org/10.1109/CVPR.2012.6248021>.
- [22] Matthijs Douze, Arnau Ramisa, and Cordelia Schmid. Combining attributes and fisher vectors for efficient image retrieval. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 745–752, 2011. doi: 10.1109/CVPR.2011.5995595. URL <http://dx.doi.org/10.1109/CVPR.2011.5995595>.
- [23] Behjat Siddiquie, Rogério Schmidt Feris, and Larry S. Davis. Image ranking and retrieval based on multi-attribute queries. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 801–808, 2011. doi: 10.1109/CVPR.2011.5995329. URL <http://dx.doi.org/10.1109/CVPR.2011.5995329>.

- [24] Catherine Wah, Steve Branson, Pietro Perona, and Serge Belongie. Multiclass recognition and part localization with humans in the loop. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2524–2531, 2011. doi: 10.1109/ICCV.2011.6126539. URL <http://dx.doi.org/10.1109/ICCV.2011.6126539>.
- [25] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, pages 438–451, 2010. doi: 10.1007/978-3-642-15561-1\_32. URL [http://dx.doi.org/10.1007/978-3-642-15561-1\\_32](http://dx.doi.org/10.1007/978-3-642-15561-1_32).
- [26] Sukrit Shankar, Vikas K. Garg, and Roberto Cipolla. DEEP-CARVING: discovering visual attributes by carving deep neural nets. *CoRR*, abs/1504.04871, 2015. URL <http://arxiv.org/abs/1504.04871>.
- [27] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014. URL <http://arxiv.org/abs/1403.6382>.
- [28] Chao-Yeh Chen and Kristen Grauman. Inferring analogous attributes. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 200–207, 2014. doi: 10.1109/CVPR.2014.33. URL <http://dx.doi.org/10.1109/CVPR.2014.33>.
- [29] Dinesh Jayaraman, Fei Sha, and Kristen Grauman. Decorrelating semantic visual attributes by resisting the urge to share. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 1629–1636, 2014. doi: 10.1109/CVPR.2014.211. URL <http://dx.doi.org/10.1109/CVPR.2014.211>.
- [30] Sagnik Dhar, Vicente Ordonez, and Tamara L. Berg. High level describable attributes for predicting aesthetics and interestingness. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 1657–1664, 2011. doi: 10.

1109/CVPR.2011.5995467. URL <http://dx.doi.org/10.1109/CVPR.2011.5995467>.

- [31] Rogerio Feris, Behjat Siddiquie, Yun Zhai, James Petterson, Lisa M. G. Brown, and Sharath Pankanti. Attribute-based vehicle search in crowded surveillance videos. In *Proceedings of the 1st International Conference on Multimedia Retrieval, ICMR 2011, Trento, Italy, April 18 - 20, 2011*, page 18, 2011. doi: 10.1145/1991996.1992014. URL <http://doi.acm.org/10.1145/1991996.1992014>.
- [32] Babak Saleh, Ali Farhadi, and Ahmed M. Elgammal. Object-centric anomaly detection by attribute-based reasoning. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 787–794, 2013. doi: 10.1109/CVPR.2013.107. URL <http://dx.doi.org/10.1109/CVPR.2013.107>.
- [33] Adriana Kovashka and Kristen Grauman. Attribute pivots for guiding relevance feedback in image search. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 297–304, 2013. doi: 10.1109/ICCV.2013.44. URL <http://dx.doi.org/10.1109/ICCV.2013.44>.
- [34] Adriana Kovashka and Kristen Grauman. Attribute adaptation for personalized image search. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 3432–3439, 2013. doi: 10.1109/ICCV.2013.426. URL <http://dx.doi.org/10.1109/ICCV.2013.426>.
- [35] Naman Turakhia and Devi Parikh. Attribute dominance: What pops out? In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 1225–1232, 2013. doi: 10.1109/ICCV.2013.155. URL <http://dx.doi.org/10.1109/ICCV.2013.155>.
- [36] Gordon Christie, Amar Parkash, Ujwal Krothapalli, and Devi Parikh. Predicting user annoyance using visual attributes. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28,*

- 2014, pages 3630–3637, 2014. doi: 10.1109/CVPR.2014.464. URL <http://dx.doi.org/10.1109/CVPR.2014.464>.
- [37] Adriana Kovashka and Kristen Grauman. Discovering attribute shades of meaning with the crowd. *CoRR*, abs/1505.04117, 2015. URL <http://arxiv.org/abs/1505.04117>.
- [38] David G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999. URL <http://computer.org/proceedings/iccv/0164/vol%202/01641150abs.htm>.
- [39] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010. URL [\[url\]http://www.science.uva.nl/research/publications/2010/vandeSandeTPAMI2010\[/url\]](http://www.science.uva.nl/research/publications/2010/vandeSandeTPAMI2010).
- [40] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, 8-14 December 2001, Kauai, HI, USA*, pages 511–518, 2001. doi: 10.1109/CVPR.2001.990517. URL <http://doi.ieeecomputersociety.org/10.1109/CVPR.2001.990517>.
- [41] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996. doi: 10.1016/0031-3203(95)00067-4. URL [http://dx.doi.org/10.1016/0031-3203\(95\)00067-4](http://dx.doi.org/10.1016/0031-3203(95)00067-4).
- [42] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001. doi: 10.1023/A:1011139631724. URL <http://dx.doi.org/10.1023/A:1011139631724>.
- [43] Herbert Bay, Tinne Tuytelaars, and Luc J. Van Gool. SURF: speeded up robust features. In *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part I*, pages 404–417,

2006. doi: 10.1007/11744023\_32. URL [http://dx.doi.org/10.1007/11744023\\_32](http://dx.doi.org/10.1007/11744023_32).
- [44] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 886–893, 2005. doi: 10.1109/CVPR.2005.177. URL <http://dx.doi.org/10.1109/CVPR.2005.177>.
- [45] Manik Varma and Andrew Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1-2):61–81, 2005. doi: 10.1007/s11263-005-4635-4. URL <http://dx.doi.org/10.1007/s11263-005-4635-4>.
- [46] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002. doi: 10.1109/TPAMI.2002.1017623. URL <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2002.1017623>.
- [47] Alaa E. Abdel-Hakim and Aly A. Farag. CSIFT: A SIFT descriptor with color invariant characteristics. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 1978–1983, 2006. doi: 10.1109/CVPR.2006.95. URL <http://dx.doi.org/10.1109/CVPR.2006.95>.
- [48] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. *CoRR*, abs/1412.4564, 2014.
- [49] Andrea Vedaldi, Varun Gulshan, Manik Varma, and Andrew Zisserman. Multiple kernels for object detection. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 606–613, 2009. doi: 10.1109/ICCV.2009.5459183. URL <http://dx.doi.org/10.1109/ICCV.2009.5459183>.
- [50] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Empowering visual categorization with the gpu. *IEEE Transactions on Multimedia*, 13(1):

60–70, 2011. URL [url]<http://www.science.uva.nl/research/publications/2011/vandeSandeITM2011>[/url].

- [51] Tamara L. Berg, Alexander C. Berg, and Jonathan Shih. Automatic attribute discovery and characterization from noisy web data. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I*, pages 663–676, 2010. doi: 10.1007/978-3-642-15549-9\_48. URL [http://dx.doi.org/10.1007/978-3-642-15549-9\\_48](http://dx.doi.org/10.1007/978-3-642-15549-9_48).
- [52] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. doi: 10.1145/1390681.1442794. URL <http://doi.acm.org/10.1145/1390681.1442794>.

## CURRICULUM VITAE

### Credentials

Name, Surname: Emine Gül DANACI

Place of Birth: Ankara

Marital Status: Single

E-mail: emineguldanaci@gmail.com

Address: Computer Engineering Department of Hacettepe University Beytepe, ANKARA

### Education

BSc. Hacettepe University, Computer Engineering, 2012

MSc. Hacettepe University, Computer Engineering, -

### Foreign Languages

English

### Work Experience

Software Engineer, TÜBİTAK BİLGEM Software Technologies Research Institute, 2012-

Intern, TÜBİTAK UZAY Space Technologies Research Institute, 2011

Intern, STM Savunma Teknolojileri Mühendislik ve Ticaret A.Ş., 2010

### Areas of Experiences

Image Processing, Computer Vision

### Projects and Budgets

—

### Publications

Görsel Nitelik Öğrenmede Alt-Düzye Özniteliklerin Karşılaştırılması, Sinyal İşleme Konferansı, May, 2013 Emine Gül DANACI, Nazlı İKİZLER CİNBİŞ

### Oral and Poster Presentations

—