PROJECT REPORT

CMPE 492

Title: Identification of Properties of Architectures Using Machine Learning Techniques

Advisor:

Tunga Güngör

Mehmet Doğan

mehmet.dogan1@boun.edu.tr

Sinem Dalkılıç

sinemdalkilic@gmail.com

Contents

1. Introduction and Motivation

1.1. Introduction

Image feature detection and object recognition have attracted a lot of attention in recent years. Many theoretically and practically simple and yet very efficient machine learning methodologies are developed in order to succeed in classification of every day data. Feature extraction algorithms attempt to afford reasonable answers for all inputs and to perform "most likely" matchings. The goal of our project is to identify architectural properties of buildings by using few of these techniques. Since structural compositions are quite complicated to classify into architectural movements and periods, it is a very challenging application of machine learning algorithms.

We collect images of buildings from different eras that have specific properties and apply an image analysis methodology. Initially, we implement Local Binary Pattern descriptor to extract important features from the image in terms of feature vectors. Through its extensions, LBP operator has been made into a really powerful measure of image texture and shows distinguished results in terms of accuracy and simplicity in empirical studies. “The LBP method has already been used in a large number of applications all over the world, including visual inspection, image retrieval, remote sensing, biomedical image analysis, face image analysis, motion analysis, environment modeling, and outdoor scene analysis.”[1] As an alternative feature extraction method, we have used Scale Invariant Feature Transform. It is a widely used computer vision method for object recognition, video tracking and feature matching. SIFT extracts keypoints from images and compute their descriptors. As its name suggests, it is invariant to scale, rotation, and illumination. Since results of SIFT are not directly applicable to our learning operator, K-means clustering and the concept of Bag of Features are implemented as well. Both methods are used to extract feature vectors from images and their outputs are analyzed to improve the classification process.

[1] Pietikäinen, Matti. "Image Analysis with Local Binary Patterns." Image Analysis Lecture Notes in Computer Science. (2005).

The feature vectors are then given to Support Vector Machine which is one of the most outstanding supervised learning algorithms. It is used to train given data and classify test examples into desired categories according to what it has learned. Architectural type and architectural era of buildings are the retrieved results from the learning model.

1.2. Motivation

People have wondered about the eras and architectural types of the buildings. Although, there exists many attempts using machine learning algorithms for controlling smart buildings, there are less works carried out to identify the eras of monuments.

An architectural style is a specific construction, characterized by its unique features. "Architectural history has dictated that there are complex interrelationships between different styles, including rebellion, special territoriality, revivals, and re-interpretations."[2] Therefore, it is difficult to strictly classify styles using a standard categorization. In this project, we aim to use computer vision and machine learning techniques to identify and classify the architectural types. The result of the project brings us one step closer to provide more information about the buildings.

2. State of Art

The field of identifying properties of buildings is a new study area and it is gaining importance in computer vision. Research conducted on this subject is very limited. Most similar studies are so far "Automatic Architectural Style Recognition"[3], "Identifying Architectural Style in 3D City Models with Support Vector Machines"[4] and "Architectural Style Classification Using Multinomial Latent Logistic Regression"[5].

---

[2] Xu, Zhe, Dacheng Tao, Ya Zhang, Junjie Wu, and Ah Chung Tsoi. "Architectural Style Classification Using Multinomial Latent Logistic Regression." Computer Vision – ECCV 2014 Lecture Notes in Computer Science (2014): 600-15. Web.

[3] Mathias, M., A. . Martinovic, J. Weissenberg, S. Haeghler, and Van Gool. *AUTOMATIC ARCHITECTURAL STYLE RECOGNITION*, 2011.

[4] Römer, Christoph, and Lutz Plümer,. *Identifying Architectural Style in 3D City Models with Support Vector Machines* (n.d.): n. pag. Oct. 2010.

5 Xu, Zhe, Dacheng Tao, Ya Zhang, Junjie Wu, and Ah Chung Tsoi. "Architectural Style Classification Using Multinomial Latent Logistic Regression." Computer Vision – ECCV 2014 Lecture Notes in Computer Science (2014): 600-15. Web.

“Automic Architectural Style Recognition” proposes a four-stage method for automatic building classification based on the architectural style. They demonstrate on three distinct architectural styles: Flemish Renaissance, Haussmannian, and Neoclassical. They use a steerable pyramid of Gabor filters, tuned to 4 scales and 8 orientations. It produces a feature vector containing 512 features. For the classification they use Support Vector Machine with a gaussian radial basis kernel function. The dataset they have used contains 1616 images in total. They resize all images to common size as 256*256 pixels. The feature descriptors that are used in this project are SIFT and SSIM. They achieved a detection rate of %77 with %29.4 false positive rate.

Secondly, the aim of “Identifying Architectural Style in 3D City Models with Support Vector Machines” is to improve low resolution 3D city models with semantic information about the architectural style of buildings. Wilhelminian-style is chosen to be identified. After discrete pre-processing, feature extraction, and feature weighting operations - to reduce the influence of insignificant or highly correlated features on clustering-, Support Vector Machine is trained. It is resulted in two clusters with correctly classified building ratios of more than 80%. Since the classifier in this project is based on Support Vector Machines it has proved that it is viable for this challenging task.

Third project introduces Multinomial Latent Logistic Regression(MLLR)- a latent variable algorithm- and compares its results to ones with Latent SVM. They conduct experiments in two clusters over a very wide dataset- 10 classes and 25 classes with nearly 65% and 45% accuracies. Consistently, MLLR outperforms LSVM and obtains the best classification results.

3. <u>Methods</u>

The goal of our project is to classify images into architectural eras. First step in this approach is to collect proper dataset. Our methodology is to apply feature extraction algorithms to image samples to retrieve characteristic information of every image. In this project we have implemented two common feature extraction methods- Local Binary Pattern and Scale Invariant Feature Transform. Then via a learning model, images are classified into architectural types with respect to their fetched features. Results of these two feature extraction algorithms are analyzed and compared thoroughly with different parameters inherent to practices. Overall, methodology can be divided into three major steps- data sampling, feature extraction and classification.

3.1. Data Collection

To study architectural styles we need to collect a large data sample. An architectural era has unique features such as pointed arches, rich sculptures and ornamented facades. We have used the dataset from the article “Architectural Style Classification Using Multinomial Latent Logistic Regression”.

The dataset includes 20 architecture styles and each class contains over 90 images. These images are not taken from the same angle and their coverage of buildings are not detailed in the same level. Therefore, quality of images are not similar. Utilized classes are

1. ArtNouveau,
2. Gothic,
3. QueenAnne,
4. International,
5. RussianRevival,
6. Deconstructivism,
7. AmericanCraftsman,
8. Novelty,
9. GreekRevival,
10. ChicagoSchool,
11. Byzantine,
12. Palladian,
13. Ancient Egyptian,
14. BeauxArt,
15. TudorRevival,
16. Baroque,
17. Postmodern,
18. Colonial,
19. ArtDeco, and
20. Georgian

Georgian

ArtNouveau

QueenAnne

Baroque

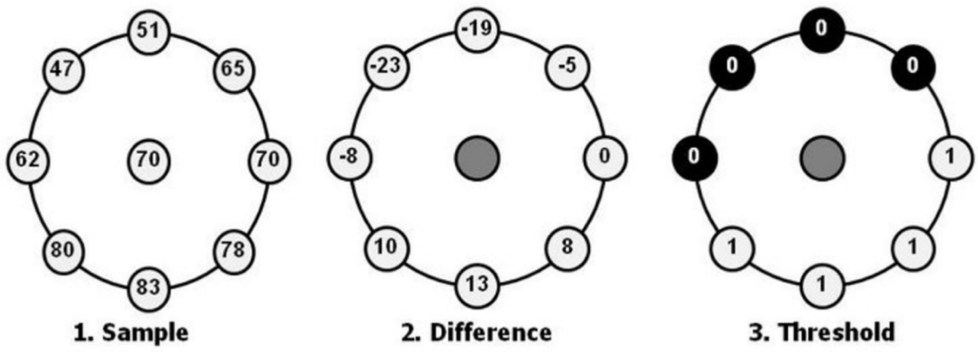
3.2. Feature Extraction

The input image contains a lot of information that majority of it is unnecessary for classification. Therefore, it is crucial to simplify the image by extracting the significant information. We have implemented two of the most widely used feature extraction methods in this project- Local Binary Pattern and Scale Invariant Feature Transform.

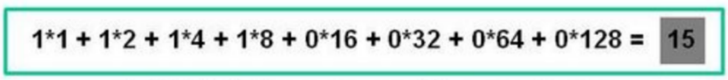
3.2.1. Local Binary Pattern (LBP)

Local Binary Pattern texture analysis operator is a grayscale texture descriptor derived from general description of the local neighborhood's texture. Main reason for why we have selected this method is that it can detect structures such as edges, lines, spots and flat areas and also nonuniform(mixed) patterns.

For each pixel on an image, we select a circularly symmetric neighborhood with radius r -surrounding the center pixel- and we decide the number of points p considered on this circle. Assume that, we have chosen values of 3 and 8 for r and p respectively for simplicity. As the initial step, an LBP value is calculated for each center pixel. The gray value of the center pixel (gc) is subtracted from the gray values of each considered point in the neighborhood (gp -gc) . If this value is greater than zero, the cell takes the value of 1, otherwise it is set to 0. An example is given below.



To calculate the LBP value of the center pixel, we can start from any neighbor cell and work in clockwise or counterclockwise. Ordering must be consistent for all pixels in all images. With 8 surrounding cells, we have 2^8=256 possible combinations of LBP values. The calculated value of the example above is



4. Multiply by powers of two and sum

The next step is to construct a histogram over the output. Since we have minimum value of 0 and maximum value of 255, we can construct a 256-bin histogram of LBP values as the feature vector with a dimensionality of 256.

However, when it is examined thoroughly, most of the combinations are categorized as *nonuniform*. A pattern is labeled as *nonuniform* if it contains more than two 0-1 or 1-0 transitions. The nonuniform patterns are assumed to be irrelevant because they do not give meaningful information about the image. Therefore, it is beneficial to reduce the dimension of the feature vector. For all *nonuniform* patterns, only one bin is reserved in the histogram. For the rest of the patterns, there is only 58 combinations which are called *uniform*. Among 58 combinations we can see that the character of these patterns can be reduced down to 9 where they are mapped to features such as flat regions, corners, lines, edges etc.

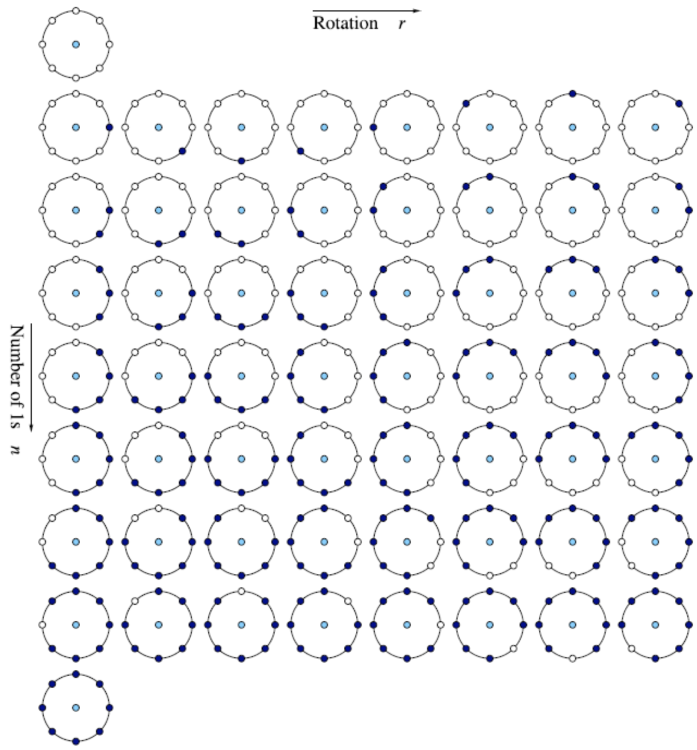
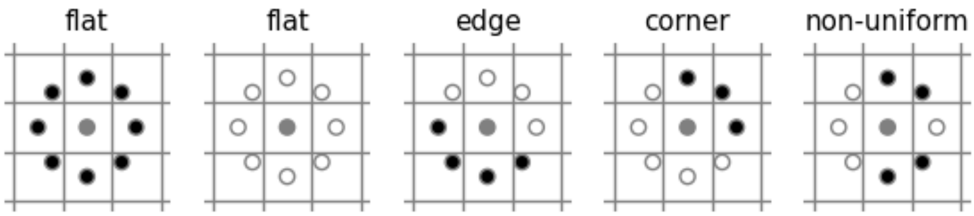


Fig. 2.4 The 58 different uniform patterns in (8, R) neighborhood



Applying uniformity is important to fulfill the rotation invariant property. Consequently, there are 9+1(10) bins in the histogram and the histogram can be used as a texture descriptor. After normalizing the histogram, we get a feature vector, which stores the frequencies of these patterns.

3.2.2. Scale Invariant Feature Transform (SIFT)

SIFT is the other algorithm that we have implemented for feature extraction. It is a widely used algorithm in computer vision to detect and describe local features in images. David Lowe defines the features in his famous paper as follows: "The features are invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection."

Computation used to generate image features can be classified in four major groups as Scale-Space Extrema Detection, Keypoint Localization, Orientation Assignment and Keypoint Detector. At the very first step, scale space of image is constructed by using different σ values with difference of gaussians. By this method potential interest points that are invariant to scale and rotation are identified. By subtracting different gaussians (i.e. blurred images with variant σ values), laplacian of gaussians is calculated. At Keypoint Localization stage, each pixel is centered and if it is a local minima or maxima, it is marked as a potential keypoint. To increase the performance of the algorithm, number of keypoints is limited by setting a threshold. Computing relative orientation and magnitude in a 16x16 neighborhood at a key point is the third stage of the method.

As the final step, keypoint region is divided into 4x4 squares and the gradients for each pixel are computed. Then for each 4x4 region, spatial weighted histogram with 8 bins is formed. Finally, these 16 histograms are concatenated and one long array containing vectors of 128 dimensions as descriptor of the key point is produced.

Keypoints Detected On a Baroque Architecture

3.2.2.1. K-means Clustering & Bag-of-Features

By applying SIFT algorithm, feature vectors are extracted from images. Since the model produces different number of keypoints, output of SIFT algorithm can not be given as input to SVM directly. It is different for each image, however SVM should be supplied with standard sized vectors. That's why the SIFT keys derived from an image are used in a nearest-neighbour approach to form a smooth input for SVM.

Because of excessive number of SIFT descriptors, we have to implement some clustering methods to bin together the descriptors that represent the same or similar features. One might think each SIFT descriptor as a “word” that when they are added up they form a vocabulary. Simply, with K-means Clustering algorithm we take each vector-with size of 128- of each image, define our range of space and divide the whole space into K clusters. As a result of this process, k bins which contain vectors from all training images are produced. Each image has different number of these vectors and all of them are utilized to define the space and its each vector is assigned to K-clusters.

Furthermore, with the concept of Bag-of-Features each image is described in terms of these bins. In other words, up to here keypoint descriptors of each image are placed into K-bins. Hence, the combination of these bins that forms an image has to be calculated in order to describe an image in terms of so called “words”. In other words, since k bins symbolize fixed number of features, each image can be stated in terms of the combination of these bins. During calculations, histograms which reflect the frequency of each bin in every image are produced.

To prevent the domination of features that has variance with orders of magnitude larger than others which may result, as a result, in making estimator unable to learn from other features, standard scalar transformation is applied to features of images. Standard scalar transform normalizes features by removing the mean and scaling to unit variance. Lastly, standardized vectors are given to SVM with its architectural class type and the machine is trained with the data.

To sum up, descriptors of key points in image are grouped into k-bins by applying K-means Clustering. Then, each image is described in terms of these bins and mean is removed for scaling features to unify variance. Finally, Support Vector Machine is trained with obtained feature vectors. The following figure shows how feature vectors extracted with SIFT are manipulated in order to train SVM.

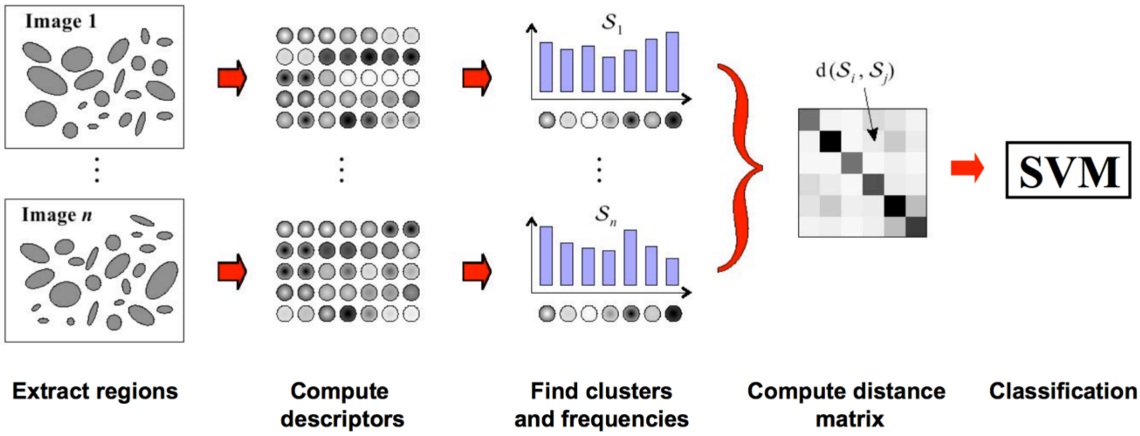
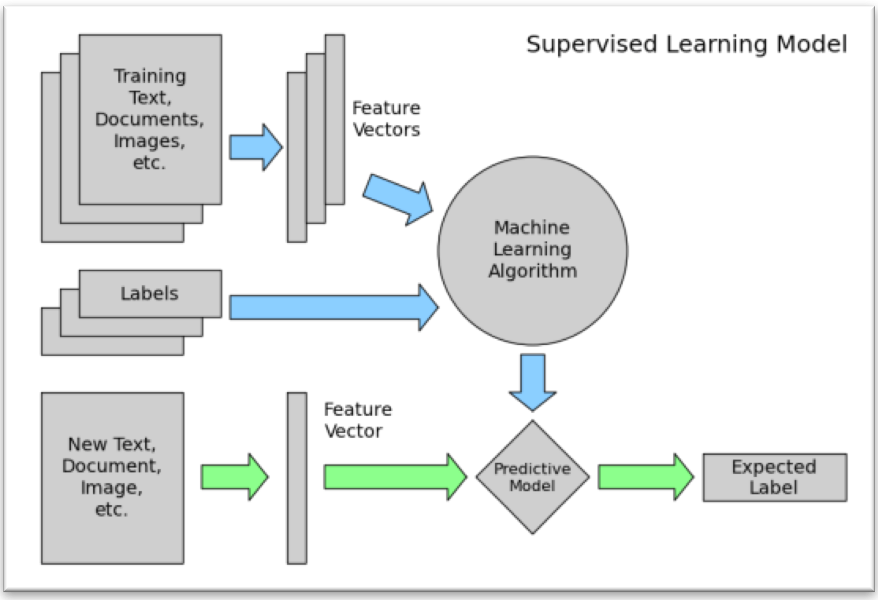


Image Classification with Bag of Features

3.2.3. <u>Training & Classification</u>

In this project Support Vector Machine(SVM) is used as a classifier. In machine learning, support vector machines are a set of supervised learning methods that are used for classification, and regression analysis. Advantages of SVM are mainly its effectiveness in high dimensional spaces and its memory efficiency since it uses a subset of training points in feature vectors.



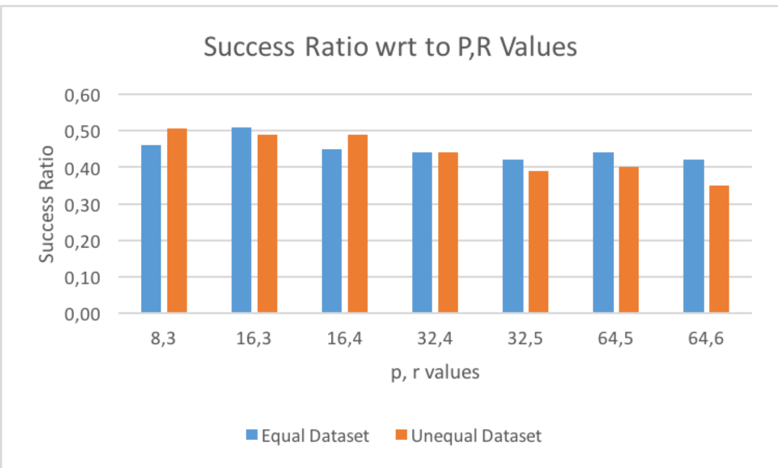
SVM training algorithm takes a set of data as training examples which are labeled with their categories. It learns from given data and constructs a model. With this model, it assigns new instances into one of these categories.

Basically, an SVM model represents these examples as points in space and they are mapped in such a way that examples of distinct categories are divided by an obvious gap that is as broad as possible. It finds the hyperplane that maximizes the margin between the positive and negative examples. In the same manner, new test examples are reflected into same space and their classes are predicted according to which side of the gap they fall.

4. <u>Results</u>

We have extracted feature vectors from two descriptors- LBP and SIFT. SVM trained the vectors and new images are introduced to test the methodology. Out of 20 classes and over 3000 images, we have implemented approximately 300 test cases. Carried out experiments are based on comparing different parameters of our method to find the best combination of parameters. Mainly, the convenience of two distinct feature extraction algorithms to our study area is tested. To manage this, each method is operated with different values of parameters. We have measured the success ratios and running times.

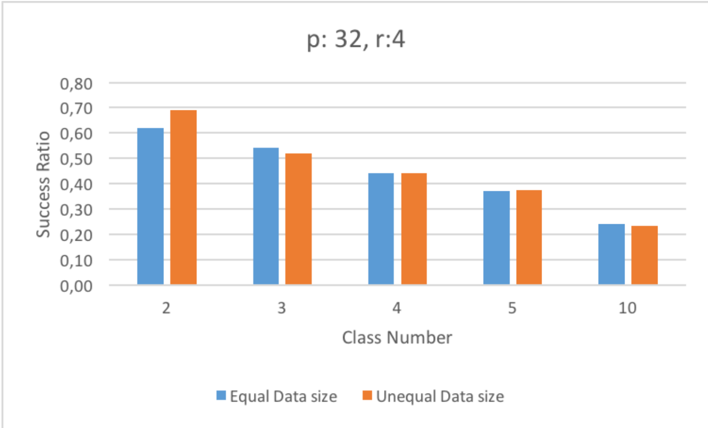
LBP is tested with four different parameters- class number, equal/unequal dataset size, p (number of points) and r (radius). Tests are run with 2, 3, 4, 5 and 10 classes. For p and r values, following pairs are used: (8,3), (16,3), (16,4), (32,4), (32,5), (64,5), and (64,6). Significant statistics are shown below.




p: 8, r:3
Success Ratio
0,00
0,10
0,20
0,30
0,40
0,50
0,60
0,70
0,80
0,90
2
3
4
5
10
Class Number
Equal Data size
Unequal Data size

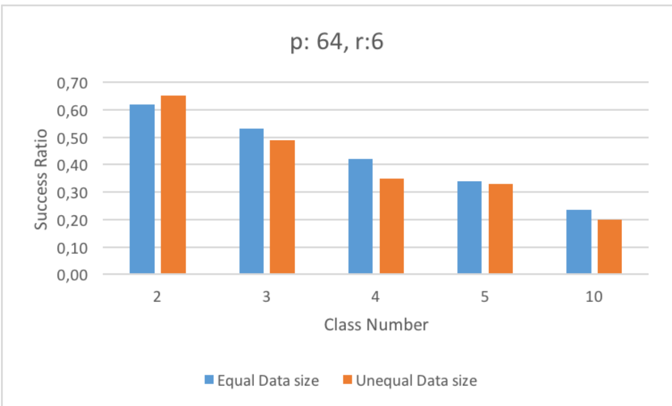

p: 16, r:3
Success Ratio
0,00
0,10
0,20
0,30
0,40
0,50
0,60
0,70
0,80
2
3
4
5
10
Class Number
Equal Data size
Unequal Data size


p: 16, r:4
Success Ratio
0,00
0,10
0,20
0,30
0,40
0,50
0,60
0,70
0,80
2
3
4
5
10
Class Number
Equal Data size
Unequal Data size

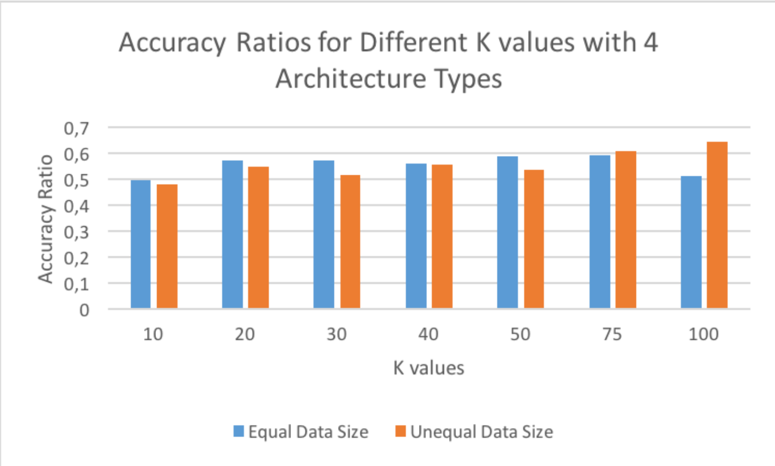

p: 32, r:4
Success Ratio
0,00
0,10
0,20
0,30
0,40
0,50
0,60
0,70
0,80
2
3
4
5
10
Class Number
Equal Data size
Unequal Data size

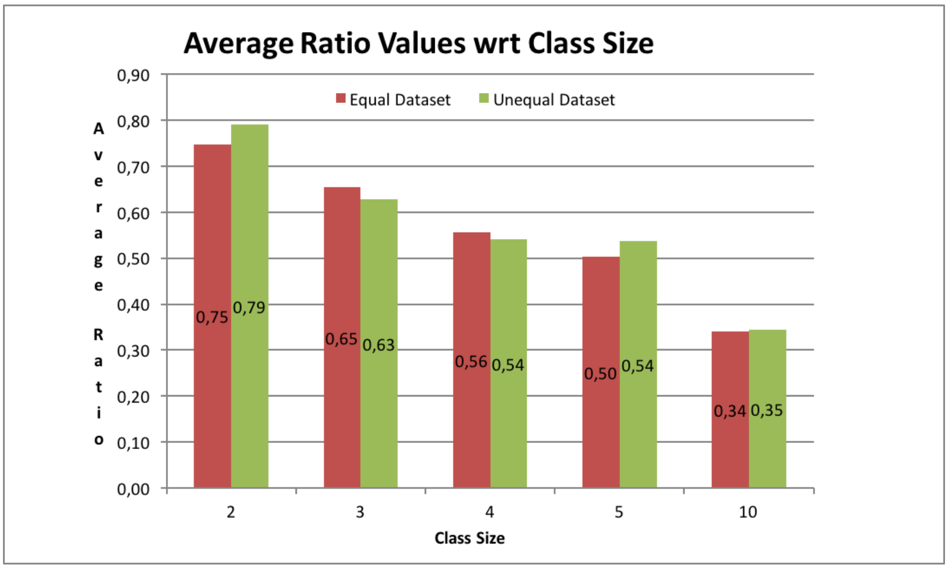

p: 32, r:5
Success Ratio
0,00
0,10
0,20
0,30
0,40
0,50
0,60
0,70
2
3
4
5
10
Class Number
Equal Data size
Unequal Data size

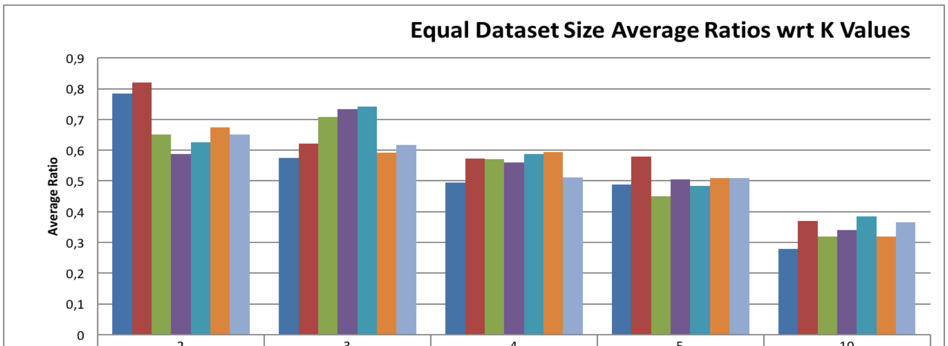

p: 64, r:5
Success Ratio
0,00
0,10
0,20
0,30
0,40
0,50
0,60
0,70
2
3
4
5
10
Class Number
Equal Data size
Unequal Data size


p: 64, r:6
Success Ratio
0,00
0,10
0,20
0,30
0,40
0,50
0,60
0,70
2
3
4
5
10
Class Number
Equal Data size
Unequal Data size

SIFT is applied with three different parameters- class number, equal/unequal dataset size, and k (cluster number). To be consistent, rests are run with 2, 3, 4, 5 and 10 classes. For k parameter 10, 20, 30, 40, 50, 75 and 100 are used. Overall results are shown in the following figures.

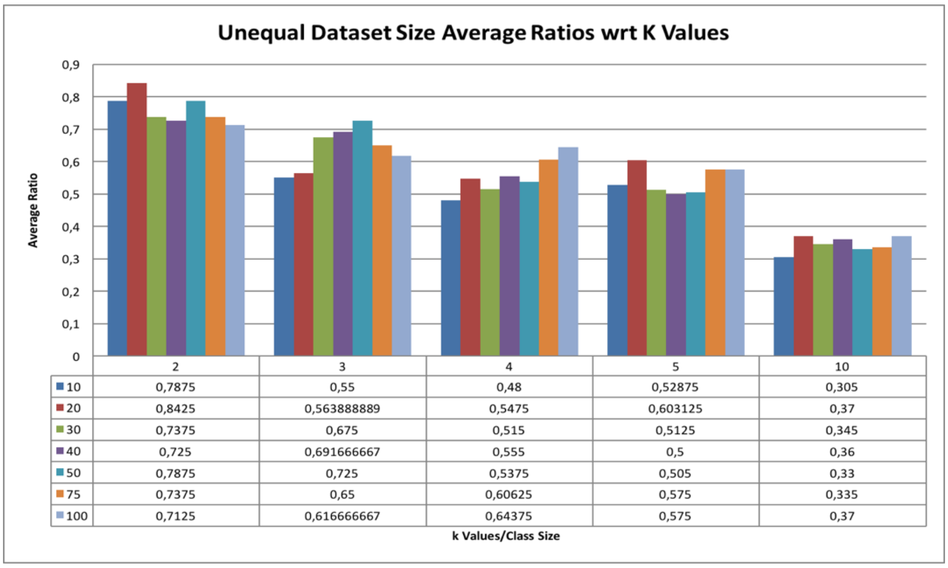






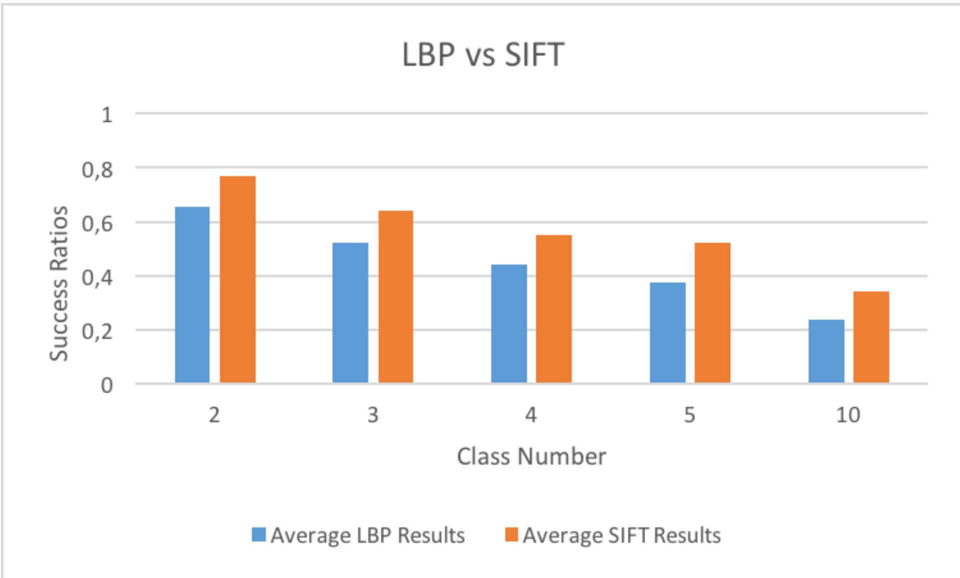
	2	3	4	5	10
10	0,785	0,575	0,495	0,4875	0,28
20	0,82	0,622222222	0,5725	0,57875	0,37
30	0,65	0,708333333	0,57	0,45	0,32
40	0,5875	0,733333333	0,56	0,505	0,34
50	0,625	0,741666667	0,5875	0,485	0,385
75	0,675	0,591666667	0,59375	0,51	0,32
100	0,65	0,616666667	0,5125	0,51	0,365

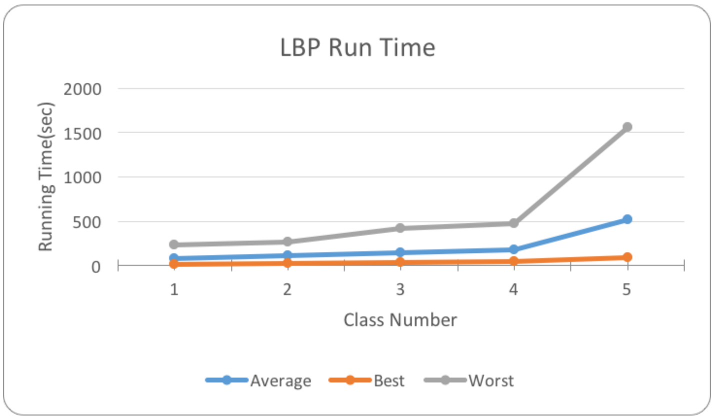
k Values/Class Size

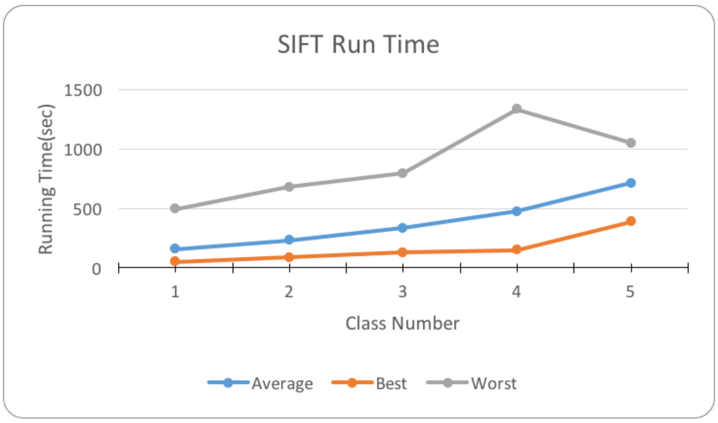


	2	3	4	5	10
10	0,7875	0,55	0,48	0,52875	0,305
20	0,8425	0,563888889	0,5475	0,603125	0,37
30	0,7375	0,675	0,515	0,5125	0,345
40	0,725	0,691666667	0,555	0,5	0,36
50	0,7875	0,725	0,5375	0,505	0,33
75	0,7375	0,65	0,60625	0,575	0,335
100	0,7125	0,616666667	0,64375	0,575	0,37

Average accuracy ratios and run times of SIFT and LBP are compared as well. Average, best and worst run times are computed for each method.







5. Conclusion and Discussion

We have implemented two techniques to identify architectural style of monuments: LBP and SIFT. As it is expected, increasing class number has resulted in decreased accuracy ratios. However, there are some exceptions. According to results, in LBP models with p value of 16 have the optimal accuracy ratio. For smaller class numbers, unequal datasets are more accurate than equal datasets. As the class number increases, their accuracy ratios become similar.

Furthermore, results of SIFT are similar for success ratios with respect to increased class number. When we compare *k* values using same class numbers, we cannot see a pattern. However, we can state that for equal size datasets 50, and for unequal datasets 100 is optimal. Using equal datasets or different number of images, is another parameter in our study. In SIFT, results differ approximately 2-4%. However, for high class numbers, unequal dataset size should be chosen.

If we compare the average success ratios of discussed algorithms, it shows that SIFT algorithm results in better accuracy for classification of architecture styles. Also, run times do differ. In worst case, it takes longer times to finish tests with LBP.

6. Future Work

This project presents methods for classification of architectures with machine learning techniques. The continuum of the project can be improved with choosing the best k for K-Means Clustering in SIFT by using statistical tests like SSE(Sum of Squared Errors) or with a higher level approach, namely G-means algorithm, which is described on “Learning the K in K-Means” paper published by Greg Hamerly, and Charles Elkan. Similar approach can be applied to Local Binary Pattern algorithm for deciding best number of points and radius.

According to tests that are conducted in this project, runtime of training model takes too long. That is another direction for future research which can be applied with Neural Network Approaches that can greatly advance the performance of these state of the art of computer vision. Algorithms in our study can not be applied on large data sets because of the restriction of CPU and memory. Whereas a project which involves training a convolutional neural network may be applied to much larger data sets.

7. References

- Chen, Rongan, Zhiyi Qu, and Jianpeng Qiu. "Enhanced Bag-of-Features Model for Image Classification." 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA) (2014): n. pag. Web.
- Clemon, Jason. "SIFT: SCALE INVARIANT FEATURE TRANSFORM BY DAVID LOWE." (n.d.): n. pag. Web. 4 May 2017.
- Hamerly, Greg, and Charles Elkan. "Learning the k in k-means." NIPS. Vol. 3. 2003.
- Hanzra, Bikramjot Singh. "Texture Matching Using Local Binary Patterns (LBP), OpenCV, Scikit-learn and Python." Hanzra Tech. N.p., n.d. Web. 21 Mar. 2017.
- London, Ian. "VBoW Pt 2 - Image Classification in Python with Visual Bag of Words (VBoW)."Ian London's Blog. N.p., n.d. Web. 2 May 2017.
- Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60.2 (2004): 91-110.
- Lowe, David G. "Object recognition from local scale-invariant features." Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Vol. 2. Ieee, 1999.
- Mathias, M., A. . Martinovic, J. Weissenberg, S. Haeghler, and Van Gool. AUTOMATIC ARCHITECTURAL STYLE RECOGNITION, 2011.
- Ojala, Timo, Matti Pietikäinen, and Topi Mäenpää. "Multiresolution Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns." N.p., n.d. Web.
- Pietikäinen, Matti. "Image Analysis with Local Binary Patterns." Image Analysis Lecture Notes in Computer Science. (2005).
- Rosebrock, Adrian. "Local Binary Patterns with Python & OpenCV." N.p., 7 Dec. 2015. Web.
- Römer, Christoph, and Lutz Plümer,. Identifying Architectural Style in 3D City Models with Support Vector Machines (n.d.): n. pag. Oct. 2010.
- "Sklearn.preprocessing.StandardScaler." Sklearn.preprocessing.StandardScaler — Scikit-learn 0.18.1 Documentation. N.p., n.d. Web. 13 Apr. 2017.
- Xu, Zhe, Dacheng Tao, Ya Zhang, Junjie Wu, and Ah Chung Tsoi. "Architectural Style Classification Using Multinomial Latent Logistic Regression." Computer Vision – ECCV 2014 Lecture Notes in Computer Science (2014): 600-15. Web.