Classification of Historic Ornaments with CNN

Sevgi Altun[1], Cem Güneş[2]

0000-0002-5872-6985[1], 0000-0003-1260-9852[2]

[1] Istanbul Technical University, Graduate School of Science, Engineering, and Technology, Department of Informatics, Architectural Design Computing, Istanbul, Turkey
[2] Istanbul Technical University, Graduate School of Science, Engineering, and Technology, Department of Informatics, Architectural Design Computing, Istanbul, Turkey

This paper is a critical assessment of an exploration of computer vision and deep learning methods in an architectural heritage context. Convolutional neural network, a type of deep learning is implemented to classify a group of Anatolian Seljuk ornamental patterns. The field of computer vision offers the potentials to assist studies in the field of architectural heritage. However, there are limited studies that combine knowledge across the two fields. One frequently studied topic is image classification based on features. In this study, we took on the task of classifying Anatolian Seljuk ornamental patterns to investigate the potential. The project focused on carved ornamental patterns on flat surfaces due to ease of data collection. The group of images is collected and arranged as two different yet related datasets. The classes are floral and geometrical, and subclasses are sparse and dense for both. Two different CNN architectures are used to train models for predictions. The process and effect of dataset creation on the implementation are explained. Results are discussed from both the technical and architectural points of view, providing a basis for further interdisciplinary studies.

Received: 20.08.2020
Accepted: 25.09.2020

Corresponding Author:
altunse19@itu.edu.tr

Altun, S., & Güneş, C. (2020). Classification of Historic Ornaments with CNN, JCoDe: Journal of Computational Design, 1(3), 115-130.

Keywords: Convolutional Neural Networks, Architectural Ornaments, Anatolian Seljuk Ornaments, Architectural Heritage, Artificial Intelligence

Evrişimli Sinirsel Ağ Kullanarak Anadolu Selçuklu Desenlerinin Sınıflandırılması

Sevgi Altun[1], Cem Güneş[2]

0000-0002-5872-6985[1], 0000-0003-1260-9852[2]

[1] İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilişim Anabilim Dalı, Mimari Tasarımda Bilişim, İstanbul, Türkiye
[2] İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilişim Anabilim Dalı, Mimari Tasarımda Bilişim, İstanbul, Türkiye

Bu metin, mimari miras bağlamında bilgisayarla görü ve derin öğrenme yöntemlerinin kullanımına ilişkin bir çalışmanın değerlendirmesidir. Bir tür derin öğrenme yöntemi olan evrişimli sinirsel ağ(CNN), Anadolu Selçuklu süs desenlerini sınıflandırma amacı ile uygulanmıştır. Bilgisayarla görü, mimari miras alanında bilgi sağlama ve çalışmalara yardımcı olma kapasitesine sahip olsa da, her iki alandaki bilgileri bir araya getiren sınırlı sayıda çalışma vardır. Mimarlık tarihini çalışmalarında sıkça karşılaşılan konulardan biri olan Anadolu Selçuklu süsleme desenlerinin sınıflandırılması, söz konusu potansiyeli araştırmak için bir örnek olarak seçilmiştir. Proje, veri toplama kolaylığı nedeniyle düz yüzeylerde oyma ile edilen süsleme desenlerine odaklanmıştır. Çalışma için kullanılacak fotoğraflar bir araya getirilmiş ve iki farklı ancak birbiriyle ilişkili veri kümesi oluşturacak şekilde işlenmiştir. Sınıflar ve alt sınıflar bitkisel (seyrek / yoğun), geometrik (seyrek / yoğun) olarak belirlenmiştir. Daha sonra derin öğrenme modellerini eğitmek ve süsleme sınıfı öngörülerini elde etmek adına iki farklı evrişimli sinirsel ağ(CNN) mimarisi kullanılmıştır. Bu çalışmanın sonuçları hem teknik hem de mimari açıdan incelenmiştir. Veri kümesi oluşturmanın hem uygulama hem süreç üzerindeki etkisi incelenmiştir. Böylece çalışma, gelecekteki kültürel miras ve yapay zeka konularında disiplinler arası araştırmalara temel oluşturmayı amaçlamıştır.

Teslim Tarihi: 20.08.2020
Kabul Tarihi: 25.09.2020

Sorumlu Yazar:
altunse19@itu.edu.tr

Altun, S., Güneş, M. C. (2020). Evrişimli Sinirsel Ağ Kullanarak Anadolu Selçuklu Desenlerinin Sınıflandırılması. JCoDe: Journal of Computational Design, 1(3), 115-130.

Anahtar Kelimeler: Evrişimli Sinirsel Ağ, Mimari Süslemeler, Anadolu Selçuklu Süslemeleri, Mimari Miras, Yapay Zeka

1. INTRODUCTION

In this paper, the results of a machine learning experiment on the classification of architectural ornaments are used as an example to discuss the process of computer vision implementation for architectural heritage studies. The importance of problem definition and datasets are highlighted. Benefits and deficiencies are critically reviewed.

Anatolian Seljuk ornamental patterns are one of the popular topics of research in both architectural design computing and the history of architecture. Beginning from the design process part and whole relations of these patterns depend on the geometrical rules and production methods. The analysis can be done focusing on various features, semantic, or physical. In the scope of this study, we focus on the basic visual aspects of the ornaments and try to classify them concerning their figure-ground relation and curviness.

We are using convolutional neural network architectures (CNN), a type of deep neural network that is mainly used for computer vision studies. Implementation of CNN for classification may provide us a beneficial analysis of the physical and semantic relations of the ornaments. Also, it may be used for defining features of architectural ornaments in different periods.

Two datasets are created for the study. For the initial experiments, a simpler dataset with 1010 images of 2 classes is created and used. The second dataset consists of 1400 images with 4 classes. The reason for the difference between numbers is the insufficient number of images acquired for the additional two classes. The experiments are done using two different CNN architectures, details of which are explained in section 5.

2. RELATED WORK

In the Anatolian Seljuk Period (1077- 1308) geometric ornaments are widely used in monumental architectures such as mosques, caravanserais, and hospital.

Studies on the classification of this type of historical ornaments focus generally on their visual characteristics. The most common classification is based on the distinction between geometric and floral patterns. Subclasses used for this kind of classification are star systems, slip layouts, badges, and domes for geometric patterns and palmed, lotus, rumi, and acanthus for the floral patterns (Algan, 2008). An alternate sub classification of the patterns that are deemed as star systems relies on symmetry operations for classification (Kaplan & Salesin, 2004). While ornaments can also be classified as unit-based and line-based by defining the continuities (Bulut, 2017), one other study classifies them based on the number of ornamental strips that are brought together (Ertunç, 2016).

Adhering to a limited number of images, we use a general and simple classification, namely the distinction of floral and geometric aspects. Even though there are no studies on figure-ground relationships of the ornaments, we also try to classify the patterns based on their figure-ground relations in subclasses in a novel attempt.

Our ultimate motivation in this is to guide future studies about classifications that are informed by the making process, materials, tools of designs, styles, and their relation to the function of structures with respect to time. It is known that master builders of the Anatolian Seljuk period traveled through cities building various structures and there are researches to detect their traces on the craftsmanship for the architecture history studies (Ödekan, 1977). Our classification study can provide clues for this kind of research since figure-ground relation is can be interpreted in relation to making methods and tools.

3. DATA PREPARATION

3.1 Source

The source of the images used in datasets is a research project supported by TÜBİTAK, “Computer-Aided Analysis of Design Processes of Two-Dimensional Geometric Patterns in Anatolian Seljuk Architecture” (114K283) coordinated by Prof. Dr. Mine Özkar, in 2014-2016. 92 images taken from the archive of the project are processed and classified by hand. The images of ornaments are from Anatolian Seljuk structures from Sivas, Kayseri, Konya, Erzurum, Amasya, Kütahya, Tokat, Erzincan.

The images taken from the source are rescaled and clipped from 3008 x 2000 pixels to 256 x 256 pixels. During this process, we zoomed into the related part of the image and centered the pattern in the square. Each image after preparation consist of only one ornamental pattern and the pattern occupies more than 80% of the image. Approximately 15 dataset images are extracted from each source image. The RGB values of the images are kept as they are at this stage.

3.2 Datasets

Two different datasets are used for the study. The first dataset consists of 1010 images in total with size 256 x 256 pixels divided into 2 classes: floral and geometrical. 10 % of these images are considered as validation data which is used for evaluation during training and another 5 % is the test data for evaluation after training. The second dataset consists of 1400 images in total with size 256 x 256 pixels divided into 4 classes: dense floral, sparse floral, dense geometrical, sparse geometrical (**Figure 1**). 10% of these images are validation data and another 10% is test data. As it is seen in the names this dataset consists of sub-categories of the first dataset (**Figure 2**).

3.3 Data Augmentation

One of the common problems encountered in machine learning studies is overfitting. When the number of images is too limited, or the range of the data is too narrow models start to remember specific examples instead of learning how to classify new images. This is called overfitting. Since we have a limited number of images, we applied data augmentation to training data to enlarge the dataset to avoid overfitting problems. Some transformations such as shear, zoom, rotation, width shift, height shift, vertical flip, horizontal flip are applied to the images. These types of transformation methods are compatible with ornament creation methods as they also use flipping, axial symmetries, and rotations (Kaplan & Salesin, 2004). In addition to translations, images are converted from RGB to grayscale. Since only stone ornaments are used within the scope of this project and in most of the examples, these show similar characteristics, color is not one of the identifiers. The sizes of the input images are reduced from 256 x 256 x 3 pixels to 224 ×224 ×1 pixel since this is the procedure that was followed in AlexNet architecture. Smaller sized images reduce the number of parameters and helps to save GPU power. Validation data sizes are also reduced to 224×224×1 pixel. Both training and validation data are arranged in random batches of 64 images.

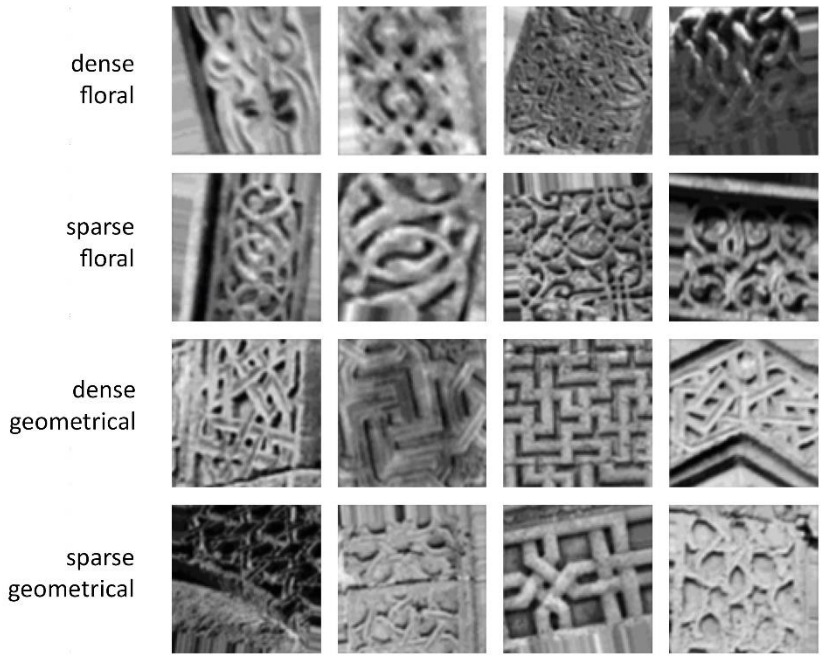

Figure 1: Data augmentation applied samples from the dataset (224 x 224 x 1 pixels)

3.4 Normalization

RGB values of input images are rescaled from between the range of 0 to 255 to between 0 and 1 to keep pixel values always in the dynamic range of the data type. This makes our model consistent and helps us avoid any data corruption by keeping the results of calculations in the computable interval.

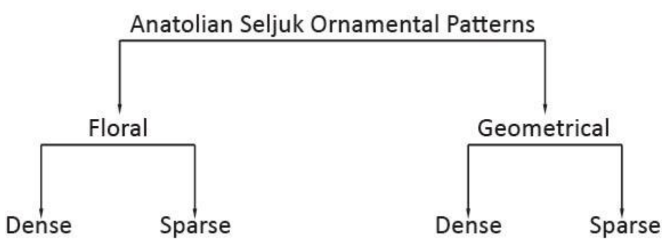

Figure 2: Class structure of the dataset

4. ENVIRONMENT

The working environment for the project is Google Colab. It is used for implementations and as a coding environment. It provided a free and powerful GPU for us. It enables us to work together on files and share easily. TensorFlow Keras open-source framework is used. The dataset is saved as .jpeg files to folders considering their classes and imported from Google Drive into Google Colab.

5. CNN ARCHITECTURES

We experimented with two different architectures for the project. One based on AlexNet architecture and the other is based on ResNet architecture (**Figure 3-4**).

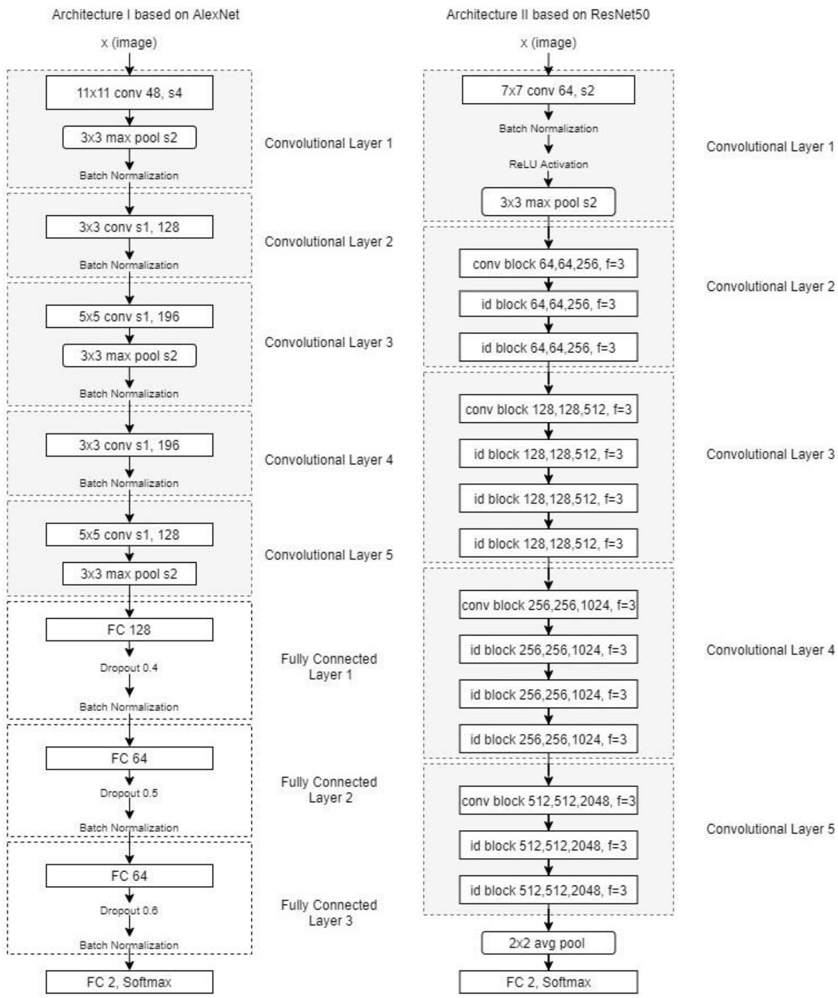

Figure 3: Details of CNN architectures

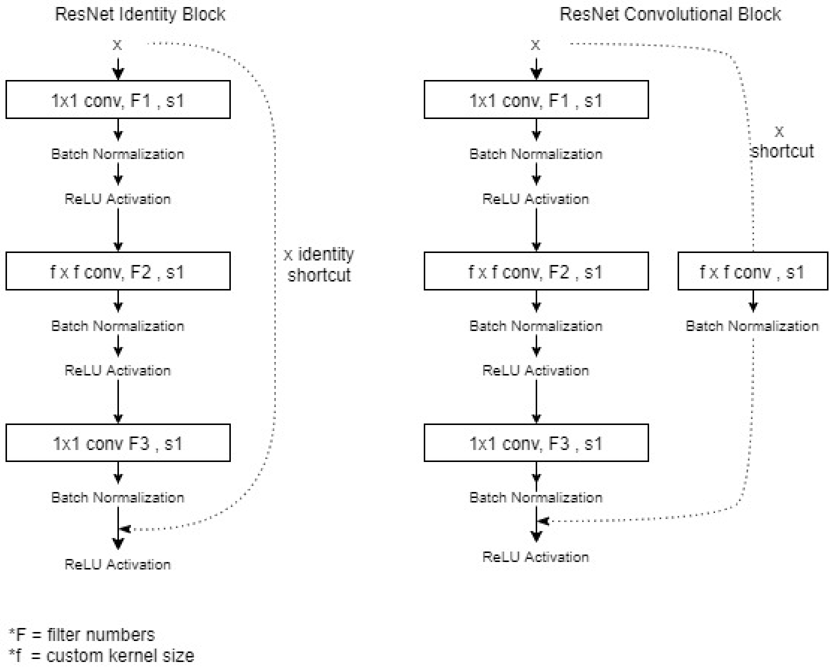

Figure 4: Details of ResNet architecture blocks

5.1 CNN Architecture I

The first one of the architectures is based on AlexNet architecture which is the primary successful CNN implementation example in computer vision studies. (Krizhevsky et al., 2012).

5.1.1 Overall Architecture

The first convolutional layer filters the 224×224×1 pixels input image with 48 kernels that are each the size of 11×11 with the stride of 4×4, then we apply overlapped max pooling and batch normalization. Kernels in other words filters are the feature extractors of given input (**Figure 5**). They are unique square-based-arrays which can process different characteristics of the image on the other hand stride (s) is a value of the shifting of the convolution or pooling filter over the n-dimensional array after each calculation step. Max pooling is a method to extract the maximum values of initial inputs in the pooling filter zone and map them in smaller sized array (**Figure 6**). This process reduces the computational cost by decreasing parameter numbers. Batch Normalization (BN) is a normalization method which organizes the data to be easily processable without losing information by scaling it between 0 and 1. Mean and variance values of mini batches are used to create functional distribution after normalization. It helps to prevent overfitting and make the networks more stable (Ioffe & Szegedy, 2015). The second convolutional layer filters the 27×27×48 pixels input with

128 kernels the size of 3×3 with the stride of 1×1. After batch normalization, the third layer takes the 27×27×128 pixels input and filters with 196 kernels the size of 5×5 with the stride of 1×1. We applied max pooling and batch normalization. The fourth convolutional layer takes input 13×13×196 pixels and filters it with 196 kernels with the size of 3×3 and stride 1×1. We applied batch normalization. The fifth and final convolutional layer takes the 13×13×196 pixels input and filters with 128 kernels with the size of 3×3 and stride of 1×1. Max pooling and batch normalization are followed by a Gaussian noise layer. Three fully connected layers have 128, 64, 64 filters and 1e-4, 1e-5, 1e-5 regularization in given order. Dropout, batch normalization is applied after each fully connected layer and Gaussian noise is applied in between fully connected layers. Softmax loss which is a classifier returns probabilities of the class predictions of given image and suitable for classification of multiple images is used for the output. ReLU (Rectifier Linear Unit) activation is an activation function which prevents vanishing gradient problem and has low computational cost. Basically, based on f(x) = max (0, x) function permits positive values to next calculation step. (Glorot, Bordes & Bengio, 2011). It is used for every convolutional and fully connected layer. Adam update which is a state-of-art stochastic optimizer and categorical cross-entropy are used (Kingma & Ba, 2015).

5.1.2 Reducing the Overfitting

Since the size of our dataset is small, there was no need for a deep network. So, we reduced the filter sizes in both convolutional and fully connected layers. We tried various learning rates to fine-tune. The best performance is obtained with a 0.00005 learning rate. Dropout is the method for dropping some connections randomly between layers to intensify the learning process over different connections and promote different neurons to learn. We applied dropout with various possibilities to find the least overfitting solutions which are 0.4, 0.5, 0.6 for each dropout layer in the given order. Overlapped max pooling is applied to reduce overfitting with the pool size 3×3 with stride 2×2. Gaussian Noise is used before passing fully connected nets to reduce overfitting. 'He' is used as an initializer for the convolutional layers.

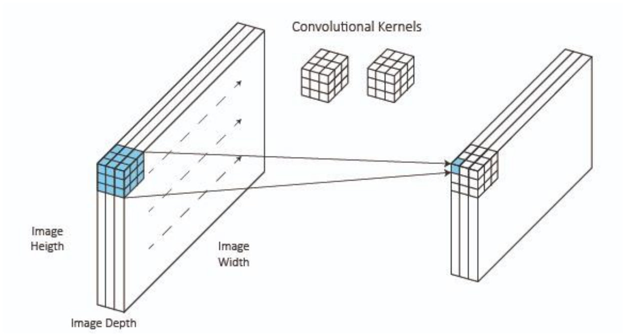

Figure 5: Visualization of convolution with the use of kernels

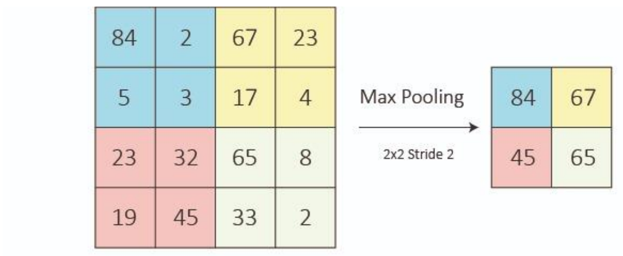

Figure 6: Visualization of max pooling

5.2 CNN Architecture II

The second architecture used for the project is mainly based on ResNet architecture (He et al., 2015). ResNet is the state of art convolutional neural network architecture which uses residual blocks to obtain better accuracy with its deep structure.

5.2.1 Overall Architecture

The architecture based on ResNet50 consists of two main blocks, namely the identity block and the convolutional block. The architecture consists of a series of ordered procedures: a convolutional layer, a batch normalization layer, ReLU activation, max pooling, second convolutional block, 2 identity blocks, third convolutional block, 3 identity blocks, fourth convolutional block, 5 identity blocks, 50 convolutional blocks, 2 identity blocks, an average pooling layer and an output layer with Softmax loss in the given order. Mainly the default values of ResNet50 are used for the architecture. Adam update and categorical cross-entropy are used.

5.2.2 Reducing the Overfitting

Because of the size of the dataset, a deeper network which has more filters resulted in overfitting. We reduced the filter sizes to ¼ of the default to prevent overfitting. We tried various learning rates to fine-tune the model, best performance is obtained with a 0.0001 learning rate. We applied dropout before the output layer to reduce overfitting, but it resulted in a worse case of overfitting. 'Xavier' is used as an initializer for the convolutional layers.

6. RESULTS

The results of the study provide both architectural and technical insights for the implementation of CNN in an architectural context (Figure 7). The classification of ornaments as floral and geometrical seem to provide accurate results. On the other hand, classification based on their figure-ground relations is more complicated and not well defined enough. Technical results are strongly related to the quality of the definition of the dataset classes.

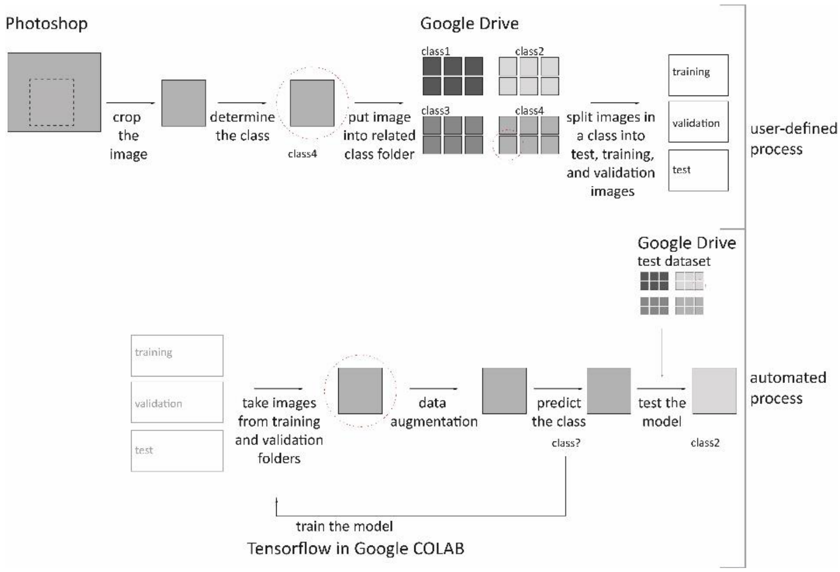

Figure 7: Diagram of the process

6.1 CNN Architecture I

During the training process, we saw that the training accuracy reached 0.9639 (high) over 1 and validation accuracy was reached 0.975 (high). Validation loss started with the value 1.52 and dropped to 0.61, training loss commenced at 2.24 and dropped to 0.34. Still, particular values show that the results are acceptable and successful; we see fluctuation and unstable results on accuracy and loss graphs (Figure 8). We tried to eliminate this type of result by canceling dropout and tuning the hyperparameters. However, we have not been able to make our model perform better. On the other hand, in the case with 4-classes, we encountered an overfitting problem. Even though we tried several options for fine-tuning we got a minimum %15 percent difference between the validation accuracy and the training accuracy. The test accuracy is 0.90 and the test loss is 0.21. These results are compatible with our training results.

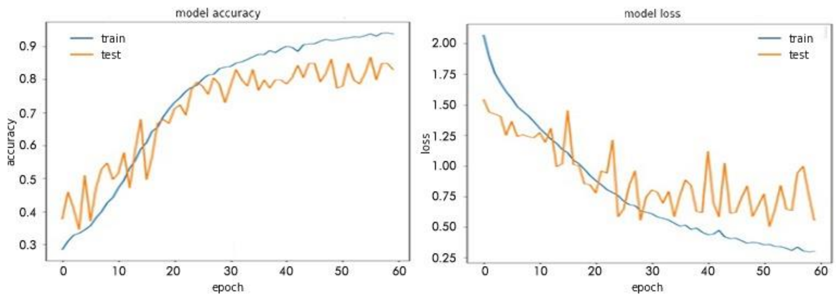

Figure 8: The training and the validation accuracies & the training and the validation losses graph of the Architecture I trained with the dataset of 2 classes.

6.2 CNN Architecture II

Training process with ResNet based architecture of 2 class dataset reached 0.99 over 1 (high) training accuracy and 0.94 (high) validation accuracy. Training loss started at 0.93 and validation loss commenced at 1.82. These respectively descended to 0.15 and 0.25. Even so, the results are acceptable for classification. The values are not equal, and we can observe immense movements in loss values in different training cycles. Additionally, the optimization process is applied to the model with 4 classes but because of the potential problems in the dataset, training results in a minimum %15 difference between training and validation accuracies. This difference shows a clear overfitting problem. As a result, we observed better stability of the accuracies in sequential training cycles and acceptable results in the 2-classes model but in the model with 4-classes we behold similar types of problems when we compared to AlexNet based architecture (**Figure 9**). The test accuracy is 0.90 and the test loss is 0.24. These results are compatible with our training results.

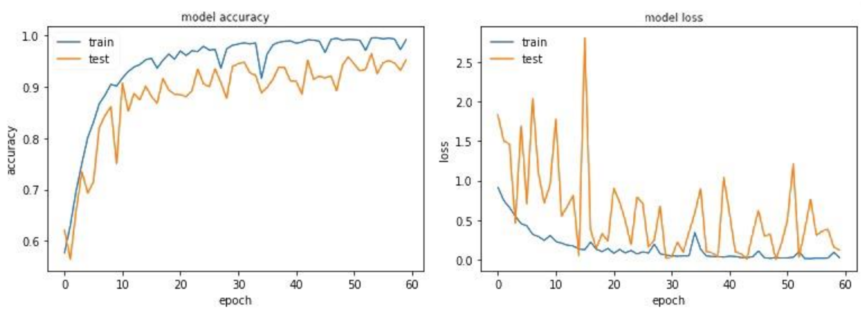

Figure 9: The training and the validation accuracies & the training and the validation losses graph of the Architecture II trained with the dataset of 2 classes.

7. CRITICAL REVIEW

Even though the results of the project are acceptable for the two classes we studied, from an architectural perspective the setup of the dataset lacks some desired features.

To start with, the ornaments which are subject to the study have many characteristics that are ignored during the classification of the dataset. The symmetry properties of ornaments, number of repetitions, location-dependent features, and material properties are some of these characteristics. These are ignored because they result in many classes, and generally validate the visual observation that each ornamental pattern is unique in some respects. Besides, while working on this kind of a comprehensive analysis and classification, the interdependencies between features can be observed. But these kinds of findings would fit the underlying purposes of the study, so their neglect is a key issue.

Another issue is that even if we classify patterns based only on the figure types and figure-ground ratios there are still subclasses and neglected properties. One of the biggest problems we encountered during classification was that the average figure-ground ratios are different for the floral and geometrical ornaments. While floral patterns are usually dense, the geometrical patterns are sparse. As a result, a ratio that can be classified as sparse in the context of floral patterns is dense in the context of geometrical patterns. This creates conflicts while deciding the classes. Another problem is when the pattern is carved into multiple layers or its main figures have grooves. The classification of these kinds of details requires deeper subclasses.

Finally, there is a possibility of human error in our classification. Figure-ground ratios of some of the patterns are close to 1 for the human eye. For that type of pattern, it is hard to decide to class for the dataset and they can create inconsistencies. Thus, even if the accuracy of CNN models is high, many features and values of individual ornaments are not represented and recognized.

8. CONCLUSION and FURTHER STUDIES

The study described here successfully classifies a group of historical architectural ornaments by using two state-of-the-art CNN architectures and two different datasets. The outputs show us using 2 classes instead of 4 gave better results due to the advantages of using more discriminative visual characteristics. The ResNet architecture was more effective and more stable compared to AlexNet due to its residual blocks.

Based on our study on the implementation of CNN in the context of architectural ornaments, we concluded that the success but acknowledge the limitations caused by the definition of the problem and the dataset. It is possible to successfully implement CNN for architectural purposes. However, it requires elaborate preliminary studies on the collection of images for setting up the dataset and the definition of the classes. Thereafter, the learning process would be easier to manipulate and the results enable us to perform comprehensive studies on architectural heritage. However, as a response to a raised number of subclasses, the computational cost would increase, and initial accuracy would get lower. This requires more work on fine-tuning of the model.

This study is an introductory example of the usage of neural networks for architectural purposes. There are various possibilities to move forward this study on architectural ornaments. As stated, only some selected properties are used for this study while there are many can be used to classify ornamental patterns and each study has potential to create information about the date, designer, location, and even semantics of the ornament. This kind of approach can also be used as the next step of the studies focusing on radial and linear symmetries of the ornaments. The classification based on the number of symmetry axes would reveal deeper information about the ornament. Another approach can be working on the number of corners and concave or convex properties of the geometry. In addition to this, another classification based on the materials (stone, marble, wood, etc.) can be carried out after creating a dataset. Combining this information with the symmetry classification, analysis can be carried on the effect of material on design and making of 2D ornaments. These classifications could not be carried out in the scope of the course project that this paper is based on since it requires a specific dataset. For the original sources used to create the data set, it would have been ideal to have access to goal-directed photographs such as consistent lighting, angles, and distance.

Implementation of computer vision in the architectural context is an interdisciplinary area of study and it brings up new conversations between the two disciplines.

Acknowledgment

The project mentioned in this paper is a term project for spring 2019 BLG506E Computer Vision course given by Hazım Kemal Ekenel in ITU, completed by the authors.

A very special thanks to Prof. Dr. Mine Özkar for providing images for the dataset and feedback throughout this project. Also, we would like to thank Res. Assist. Begüm Hamzaoğlu and Asst. Prof. Dr. Sibel Yasemin Özgan for their valuable contribution to our research on classes.

References

Algan, N. (2008). *Anadolu Selçuklu Dönemi Mimarisi Taş Yüzey Süslemelerinin İncelenmesi ve Seramik Yorumları* (Unpublished doctoral dissertation or master's thesis). Dokuz Eylül Üniversitesi Güzel Sanatlar Enstitüsü Seramik Anasanat Dalı, İzmir, Turkey.

Bulut, M. (2017). Geometrik Sistemin Çözümlenmesi - *Selçuklu Örnekleri Üzerine Birkaç Girişim. Sanat Tarihi Dergisi. 26.* 27-44. doi: 10.29135/std.292044.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. AISTATS.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 770-778).

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ArXiv, abs/1502.03167.

Ödekan, A. (1977). *Osmanlı Öncesi Anadolu Türk Mimarisinde Mukarnaslı Portal Örtüleri*. İstanbul, Turkey: İ.T.Ü. Mimarlık Fakültesi Baskı Atölyesi.

Kaplan, C., & Salesin, D. (2004). Islamic Star Patterns in Absolute Geometry. *ACM Trans. Graph. 23,* 97-119.

Kingma, D.P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. CoRR, arXiv: 1412.6980.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. CACM.

Önkol Ertunç, Ç., (2016). Anadolu Selçuklu Dönemi Taçkapıları Süsleme Şeritlerinde Tezyinat. *Pamukkale Üniversitesi İlahiyat Fakültesi Dergisi, 0*(5), 114-131. Retrieved from https://dergipark.org.tr/tr/pub/pauifd/issue/17359/181261