**T.R.**

**GEBZE TECHNICAL UNIVERSITY**

**FACULTY OF ENGINEERING**

**DEPARTMENT OF COMPUTER ENGINEERING**

# ARCHITECTURAL STYLE CLASSIFICATION

EMIRKAN BURAK YILMAZ

SUPERVISOR
PROF. YUSUF SINAN AKGÜL

GEBZE
2024

**T.R.**
**GEBZE TECHNICAL UNIVERSITY**
**FACULTY OF ENGINEERING**
**COMPUTER ENGINEERING DEPARTMENT**

# ARCHITECTURAL STYLE CLASSIFICATION

**EMIRKAN BURAK YILMAZ**

SUPERVISOR
PROF. YUSUF SINAN AKGÜL

**2024**
**GEBZE**

| | |
|---|---|
| GEBZE TECHNICAL UNIVERSITY | GRADUATION PROJECT<br>JURY APPROVAL FORM |

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 25/01/2024 by the following jury.

**JURY**

Member
(Supervisor) : Prof. Yusuf Sinan Akgül

Member : Dr. Yakup Genç

# ABSTRACT

This study presents a versatile approach integrating various neural network architectures with a focus on classifying architectural works. To address the lack of suitable datasets in the literature, a custom dataset has been created and made publicly available. The study aims to determine the most effective model, taking into account fine details in architectural styles. In this context, a comparative analysis has been conducted on four different convolutional neural network (CNN) architectures, including a baseline model trained from scratch and models using transfer learning methods with VGG, ResNet, and EfficientNet architectures. Through experiments, the EfficientNet architecture was fine-tuned, achieving an accuracy of %84.65 for 3 architects and %74.08 for 16 architects. Additionally, the two obtained models were used as feature extractors to visualize relationships among architects in a 2D space using t-SNE dimension reduction technique. These promising results indicate that these techniques can significantly contribute to architectural style analysis and serve as valuable tools for creating innovative designs through the use of creative artificial intelligence.

**Keywords:** image classification, deep learning, convolutional neural network, architectural style.

# ÖZET

Bu çalışma, mimari eserlerin sınıflandırılmasına odaklanarak çeşitli sinir ağı mimarilerini entegre eden çok yönlü bir yaklaşım sunmaktadır. Literatürdeki veri seti eksikliğini karşılamak amacıyla özel bir veri seti oluşturulmuştur. Çalışma, mimari tarzlardaki ince detayları göz önünde bulundurarak, en etkili modeli belirlemeyi hedeflemektedir. Bu bağlamda, dört farklı evrişimli sinir ağı (CNN) mimarisi üzerinde yapılan karşılaştırmalı analiz, sıfırdan eğitilen temel bir modeli ve transfer öğrenme yöntemleriyle eğitilen VGG, ResNet ve EfficientNet mimarilerini içermektedir. Deneyler sonucunda EfficientNet mimarisi fine-tune edilerek 3 mimar için %84,65 ve 16 mimar için %74,08'lik bir doğruluk elde edilmiştir. Ayrıca, elde edilen iki model, mimarlar arasındaki ilişkileri t-SNE boyut indirgeme tekniği ile 2 boyutlu uzayda görselleştirmek için özellik çıkarıcı olarak kullanılmıştır. Bu umut verici sonuçlar, bu tekniklerin mimari tarz analizine önemli katkılar sağlayabileceğini ve yaratıcı yapay zeka kullanımı ile yeni tasarımların oluşturulmasında değerli bir araç olabileceğini göstermektedir

**Anahtar Kelimeler:** resim sınıflandırma, derin Öğrenme, evrişimli sinir ağları, mimari stil.

# ACKNOWLEDGEMENT

I would like to express my profound gratitude to my supervisor, Prof. Dr. Yusuf Sinan Akgül, for providing invaluable guidance, unwavering support, and valuable suggestions throughout the entire duration of this research. Special appreciation goes to Dr. Yakup Genc for his insightful critiques, which significantly enriched the quality of the study. I extend my thanks to Doga Nalcı for their consistent support during this research endeavor. Finally, heartfelt appreciation is extended to my family for their continuous encouragement and understanding throughout this academic journey.

**Emirkan Burak Yılmaz**

# LIST OF SYMBOLS AND ABBREVIATIONS

| **Symbol or Abbreviation** | : | **Explanation** |
|---|---|---|
| VGG | : | Visual Geometry Group |
| ResNet | : | Residual Network |
| CNN | : | Convolutional Neural Network |
| t-SNE | : | t-distributed Stochastic Neighbor Embedding |
| ReLU | : | Rectified Linear Unit |
| Adam | : | Adaptive Moment Estimation - An optimization algorithm |
| RMSprop | : | Root Mean Squared Propagation - An optimization algorithm |

# CONTENTS

**Abstract** iv

**Özet** v

**Acknowledgement** vi

**List of Symbols and Abbreviations** vii

**Contents** ix

**List of Figures** x

**List of Tables** xi

**1 Introduction** 1

**2 Related Works** 2

**3 Proposed Approach** 3
- 3.1 Baseline Model and Transfer Learning . . . 3
- 3.2 Hyperparameter Tuning . . . 4
- 3.3 Training . . . 4
- 3.4 t-SNE Dimension Reduction and Clustering . . . 4

**4 Data Collection and Sampling** 5

**5 Results** 7
- 5.1 Architecture with Best Hyperparameters . . . 7
- 5.2 Training and Evaluation . . . 7
- 5.3 Ground Truth and Predictions . . . 10
- 5.4 Style Clustering . . . 11

**6 Conclusion and Future Work** 13

**Bibliography** 15

**CV** 16

**Appendices** **18**

# LIST OF FIGURES


4.1 Sample images from the dataset. . . . . . . . . . . . . . . . . . . . . . 5

5.1 Accuracy and loss values per train and validation. . . . . . . . . . . . . 8
5.2 Confusion matrix for Architect-Top dataset. . . . . . . . . . . . . . . . 8
5.3 Accuracy and loss values per train and validation. . . . . . . . . . . . . 9
5.4 Confusion matrix for Architect-All dataset. . . . . . . . . . . . . . . . 9
5.5 Model predictions. . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
5.6 Projected deep features of Architects-Top dataset. . . . . . . . . . . . 11
5.7 Projected deep features of Architects-All dataset. . . . . . . . . . . . 12

# LIST OF TABLES

4.1 Architect and sample size of collected photographs. . . . . . . . . . . 6

5.1 Performance of different network architectures. . . . . . . . . . . . . 7
5.2 Hyperparameter tuning search space and found values. . . . . . . . . 7
5.3 Best accuracy and loss, averaged across folds . . . . . . . . . . . . . 8

# 1. INTRODUCTION

The architecture of the late Ottoman and early Republic periods was shaped by the influence of the East-West dichotomy and national-universal conflicts [1]. During this era, the architectural language of public buildings initially diverged from classical European architecture and incorporated Ottoman and Seljuk references. Subsequently, structures reflecting the spirit of the young Republic were constructed in a modern style [2]. In the 1970s, the architecture of the late Ottoman period was termed the First National Style, while early Republic architecture was defined as the Second National Style [3]. However, it is noted that this classification is not rigid, and structures from this period may exhibit heterogeneous characteristics [3]. Therefore, examinations of the late Ottoman and early Republic periods have the potential to shed light on the hybrid design techniques of the era.

This research aims to develop a classifier capable of distinguishing architects of the late Ottoman and early Republic periods based on their original styles by extensively examining the architectural styles of this period. The method encompasses a comparative analysis ranging from creating training models from scratch using convolutional neural networks to transfer learning with pre-trained models. Beyond the dimension of classification, this study aims to demonstrate and explain the connections between architects. This additional step contributes to a deeper understanding of the architectural dialogue that took place during this critical period. The research has the potential to make a significant contribution to the field of architectural style analysis and creative artificial intelligence production.

# 2. RELATED WORKS

Xu et al. [4] curated a 25-class dataset from Wikimedia, employing HOG (Histogram of Oriented Gradients) for Multinomial Latent Logistic Regression. Their model successfully detected various architectural styles within single building images, including the prominently featured ”American Craftsman” style. Lee et al. [5], in 2015, utilized nearly 150,000 Google Street View images of Paris integrated with a real estate cadaster map to date building facades and track architectural element progression. Employing HOG descriptors, their approach identified features correlated with a building’s construction period, shedding light on architectural evolution.

In the work titled ”What Makes Paris Look Like Paris?” [6], Doersch et al. employed a nearest-neighbor technique to identify patches of architectural style characteristic of cities. Pesto’s research (2017) [7] focused on classifying American houses into five distinct architectural categories. This involved experimenting with three network types, including the development of a baseline network and exploration of ResNet-18 and ResNet-34 architectures for feature extraction. Emphasizing the significance of image size, the study achieved the highest accuracy of 79% using ResNet-34 as a feature extractor with 512x512 images.

Yoshimura et al. (2019) utilized the NASNet network architecture and CNN method to attain a 73% accuracy in classifying 34 Pritzker Architecture Prize winners [8]. Employing dimension reduction via principal component analysis (PCA), the study resulted in four distinct clusters, providing insights into architect similarities and differences.

Wang et al. (2023) implemented a pre-processing operation to select main structure candidate regions in architectural images and extract features related to architectural design [9]. Through the channel-spatial attention module, they generated an attention map emphasizing spatial features within diverse architectural elements and encapsulating the textures depicted in architectural images. Employing a dataset with 25 distinct architectural styles, the study achieved an accuracy score of 73% for 10 architectural classes and 64% for 25 architectural classes.

# 3. PROPOSED APPROACH

We compared four different CNN architectures for the purpose of classifying architects. Initially, we created our own basic network architecture and then conducted tests using transfer learning methods on VGG [10], ResNet [11], and EfficientNet [12] architectures. In the development of the models, we utilized the Tensorflow AI library, the numerical computing library NumPy, and pre-trained models including VGG16, ResNet50, and EfficientNetB0 from Keras..

## 3.1. Baseline Model and Transfer Learning

When dealing with small datasets, training CovNets can often lead to the problem of overfitting [13]. To overcome this challenge, transfer learning methods are employed, seeking solutions for a similar but distinct problem using models trained on large datasets such as ImageNet. Following this principle, we aimed to expedite model training and achieve optimal performance for our classification problem. Initially, a simple neural network architecture comprising three convolution layers was designed and used as a reference point to assess the performance of transfer learning models. Subsequently, we utilized Keras' pre-trained models on ImageNet, including VGG16, ResNet50, and EfficientNetB0, in our experiments. VGG16 is a straightforward and regular model with 16 layers, consisting of 13 convolution layers and 3 fully connected layers. ResNet50, with 50 layers and residual blocks, facilitates the training of deep networks using residual learning. EfficientNet belongs to a family of neural network architectures specifically designed for optimal performance on diverse computing resources, providing an effective scaling method balancing model size and accuracy. In our fine-tuning process, we aimed to preserve learned features by freezing the top layers of the pre-trained models and added additional layers to adapt the models for our specific classification task. This architecture included crucial steps such as data augmentation and preprocessing. To tailor the pre-trained models to our objectives, we replaced the original fully connected output layer with a global average pooling layer. Following this layer, we added a dense layer with 64 units, a dropout layer with a rate of 0.2, another dense layer with ReLU activation function, a second dropout layer with a rate of 0.2, and a final dense layer with softmax activation function to generate classification scores.

## 3.2. Hyperparameter Tuning

After selecting the optimal architecture, we employed Hyperband hyperparameter optimization [14] to determine the best hyperparameters. Hyperband is a powerful optimization technique that effectively balances random and grid searches by evaluating multiple hyperparameter sets simultaneously at different resource allocation levels. The optimized hyperparameters include batch size, dense layer width, dropout rate, and the choice of optimization algorithm (option between Adam and RMSprop).

## 3.3. Training

During the training process, we adopted two different approaches to enhance the adaptability and discriminative capabilities of our model: direct training and two-stage training. In the first approach, we conducted direct training on the Architects-Top and Architects-All datasets. In the second approach, we applied pre-training on the Architectural-Styles dataset obtained from Kaggle, which includes 25 different architectural styles [15], and then performed the second stage of training on the Architects-Top and Architects-All datasets. In this two-stage training approach, our goal was for the model to first learn various architectural styles and then be able to classify styles in the focused architectural periods.

## 3.4. t-SNE Dimension Reduction and Clustering

After completing the training, we removed the final softmax layer of the final model, transforming the images into feature vectors. This way, the final model was used as a feature extractor. Subsequently, to visualize these vectors in a 2D plot, we applied the t-SNE dimension reduction method, obtaining insights into the relationships and differences between architectural styles.

# 4. DATA COLLECTION AND SAMPLING


Figure 4.1: Sample images from the dataset.

Our research commenced by focusing on the works of 16 architects renowned during the late Ottoman and early Republic periods. A raw dataset containing images of targeted architects and their works was created. This dataset was named ”Architects-All,” and its contents are detailed in Figure 4.1. Additionally, a subset derived from this dataset, named ”Architects-Top,” was created. The purpose of this subset was to conduct a more in-depth examination of three architects: Giulio Mongeri, Architect Kemalettin, and Vedat Tek.

Table 4.1: Architect and sample size of collected photographs.

| **Architect** | **Sample Size** |
|---|---:|
| Adil Denktaş | 8 |
| Ali Talat Bey | 58 |
| Arif Hikmet Holtay | 21 |
| Arif Hikmet Koyunoğlu | 104 |
| Clemens Holzmeister | 205 |
| Ertuğrul Menteşe | 30 |
| Faruk Aysu | 7 |
| Giulio Mongeri | 294 |
| Mimar Kemalettin | 429 |
| Necmettin Emre | 117 |
| Paul Bonatz | 66 |
| Rükneddin Güney | 59 |
| Seyfi Arkan | 88 |
| Vedat Tek | 352 |
| Şekip Akalın | 36 |
| Şevki Balmumcu | 36 |

# 5. RESULTS

## 5.1. Architecture with Best Hyperparameters

EfficientNet emerged as the best network architecture, demonstrating superior performance in our experiments on architectural style classification. Detailed training results, including the network architectures used and accuracy values, are presented in Table 5.1. The hyperparameters for EfficientNetB0, along with the search spaces, can be found in Table 5.2.

Table 5.1: Performance of different network architectures.

| **Model** | **Accuracy** | **Loss** |
|---|---|---|
| Baseline | 60.47% | 0.8745 |
| VGG16 | 74.42% | 0.7154 |
| ResNet50 | 68.06% | 1.1169 |
| EfficientNetB0 | 82.33% | 0.4838 |

Table 5.2: Hyperparameter tuning search space and found values.

| **Hyperparameter** | **Search Space** | **Optimum Value** |
|---|---|---|
| Batch Size | 32, 64 | 256 |
| Dense Layer Width | 64, 128, 256 | 32 |
| Dropout Rate | 0.2, 0.3, 0.4 | 0.4 |
| Optimization | RMSprop, Adam | RMSProp |

## 5.2. Training and Evaluation

When evaluating various training methods and their effects on model accuracy, our most successful strategy was direct training for Architects-Top and two-stage training for Architects-All. Training results, summarizing the performance of different models under different strategies, can be found in Table 5.3.

Table 5.3: Best accuracy and loss, averaged across folds

| Training Approach | Model | Accuracy | Loss |
|---|---|---|---|
| Direct Traning | architects-top | 84.65 | 0.4659 |
| | architects-all | 72.25 | 0.9371 |
| Two-stage Training | architectural-styles | 66.01 | 1.1576 |
| | architects-top-finetune | 82.33 | 0.5270 |
| | architects-all-finetune | 74.08 | 0.9060 |

The model trained directly on Architects-Top demonstrated exceptional performance, achieving the highest accuracy. Comprehensive training statistics are available in Figure 5.1. Furthermore, the confusion matrix for the model is provided in Figure 5.2.

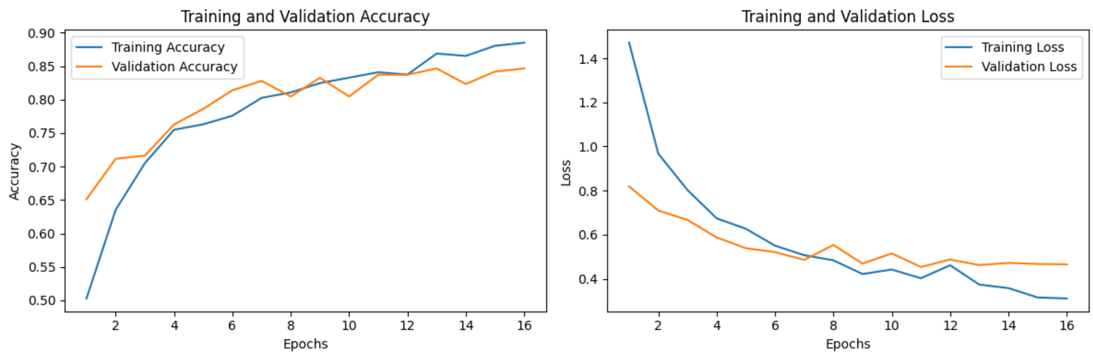


Figure 5.1: Accuracy and loss values per train and validation.

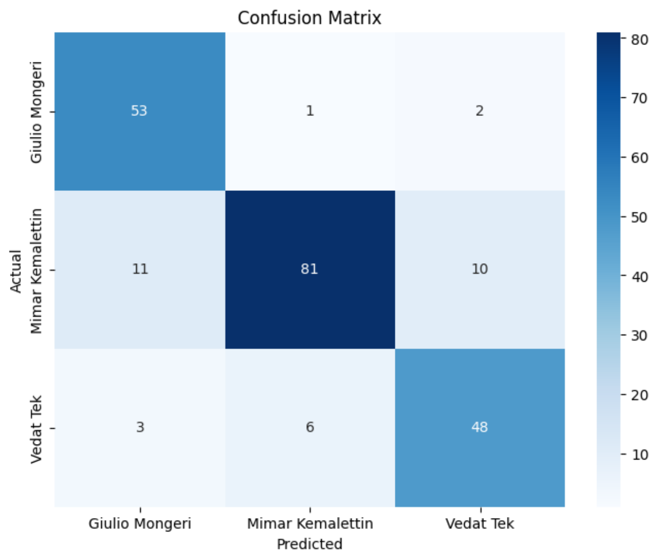


Figure 5.2: Confusion matrix for Architect-Top dataset.

On the other hand, the model trained in two stages on Architects-All exhibited slightly superior performance compared to the directly trained model. Detailed training statistics for this model are presented in Figure 5.3, encompassing accuracy and loss values for both training and validation sets. Furthermore, a confusion matrix representing the effectiveness of distinguishing architectural styles in the late Ottoman and early Republic periods in Turkey is visually depicted in Figure 5.4.

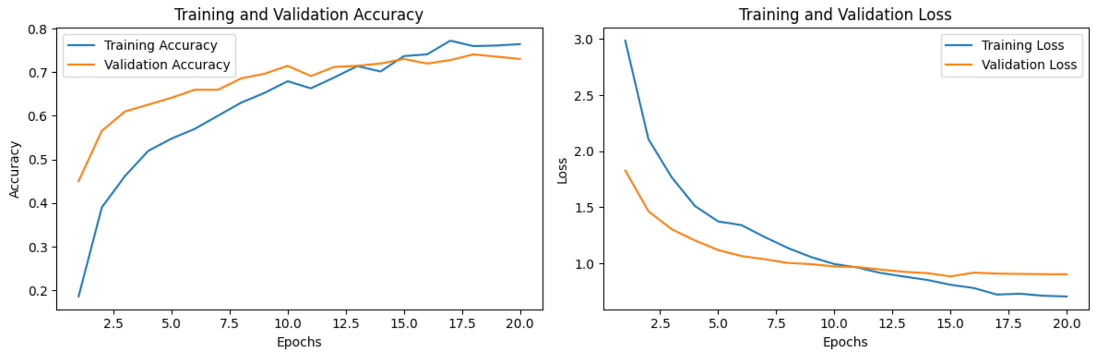


Figure 5.3: Accuracy and loss values per train and validation.

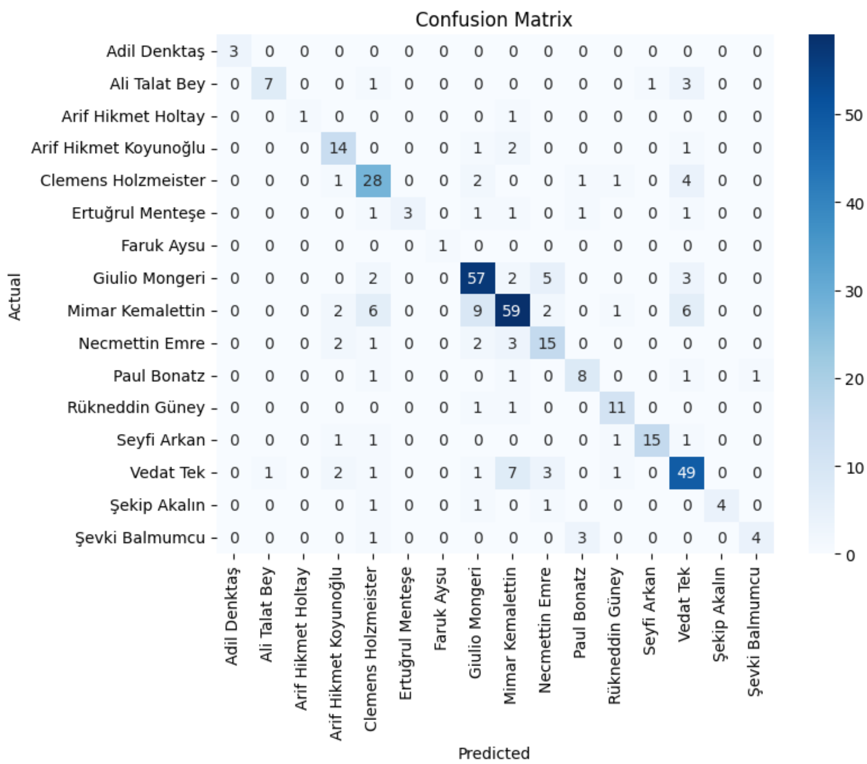


Figure 5.4: Confusion matrix for Architect-All dataset.

## 5.3. Ground Truth and Predictions

The model predictions and the corresponding ground truth for the images are presented in Figure 5.5. Instances of mismatch between the model prediction and ground truth are highlighted in red.

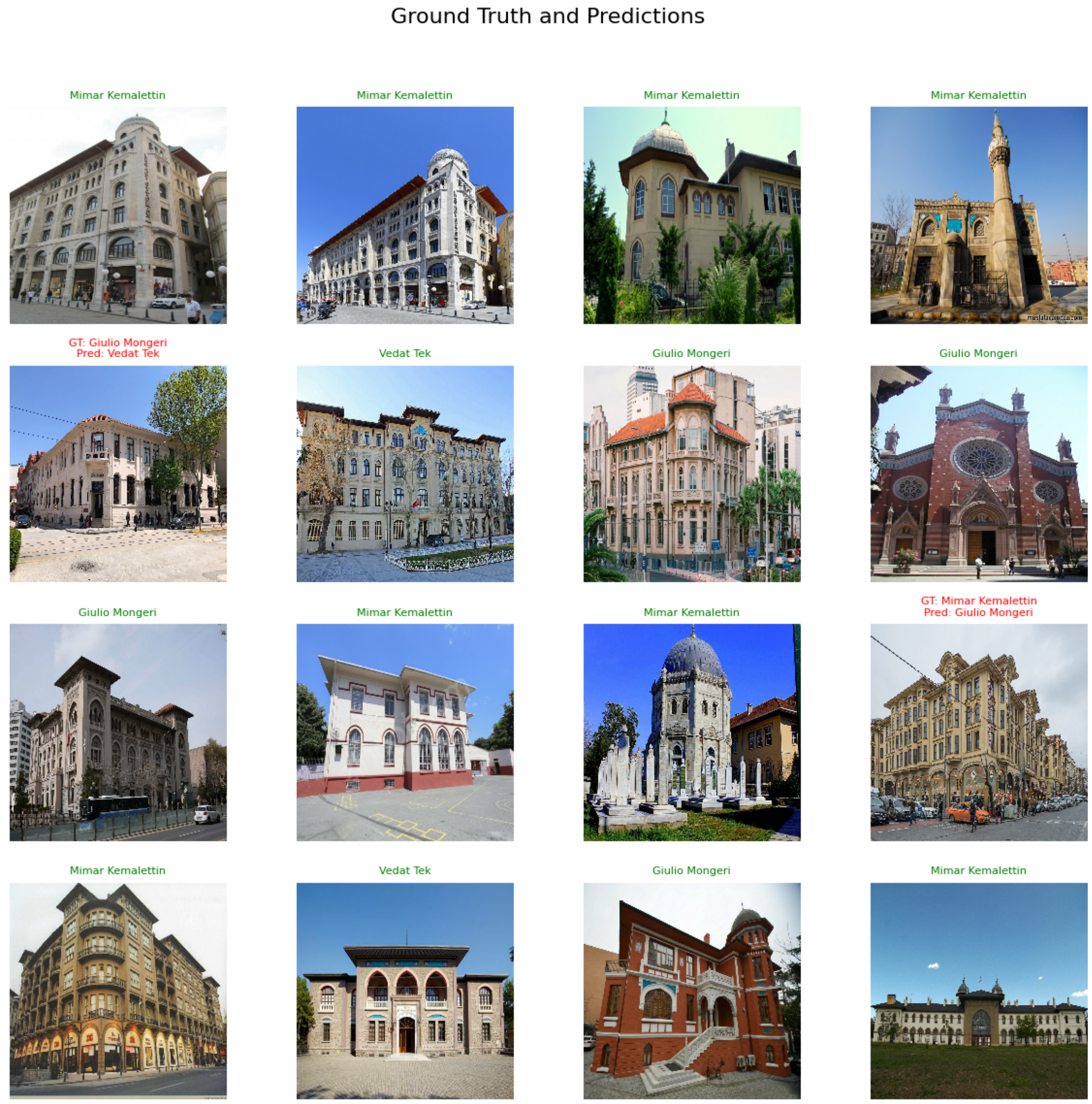


Figure 5.5: Model predictions.

## 5.4. Style Clustering

The clustering results for the Architects-Top dataset are presented in Figure 5.6, where three architects converge towards the center of the plot. Particularly, it is observed that Architect Kemaleddin is positioned between Gulio Mongeri and Vedat Tek, indicating an influence from both architectural styles.

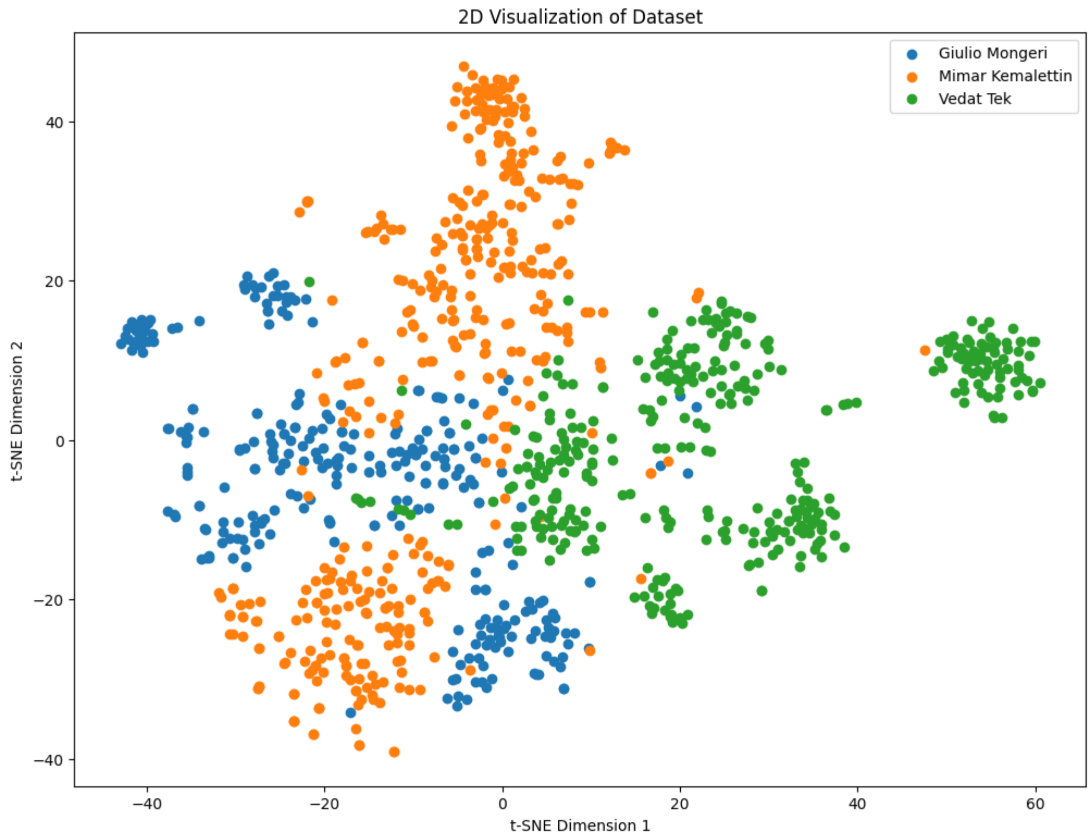


Figure 5.6: Projected deep features of Architects-Top dataset.

The clustering results for the Architects-All dataset are presented in Figure 5.7. Particularly, it is observed that Vedat Tek exhibits works in notably different styles. Additionally, an interesting similarity is observed between the architectural styles of Necmettin Emre, Architect Kemalettin, and Vedat Tek.

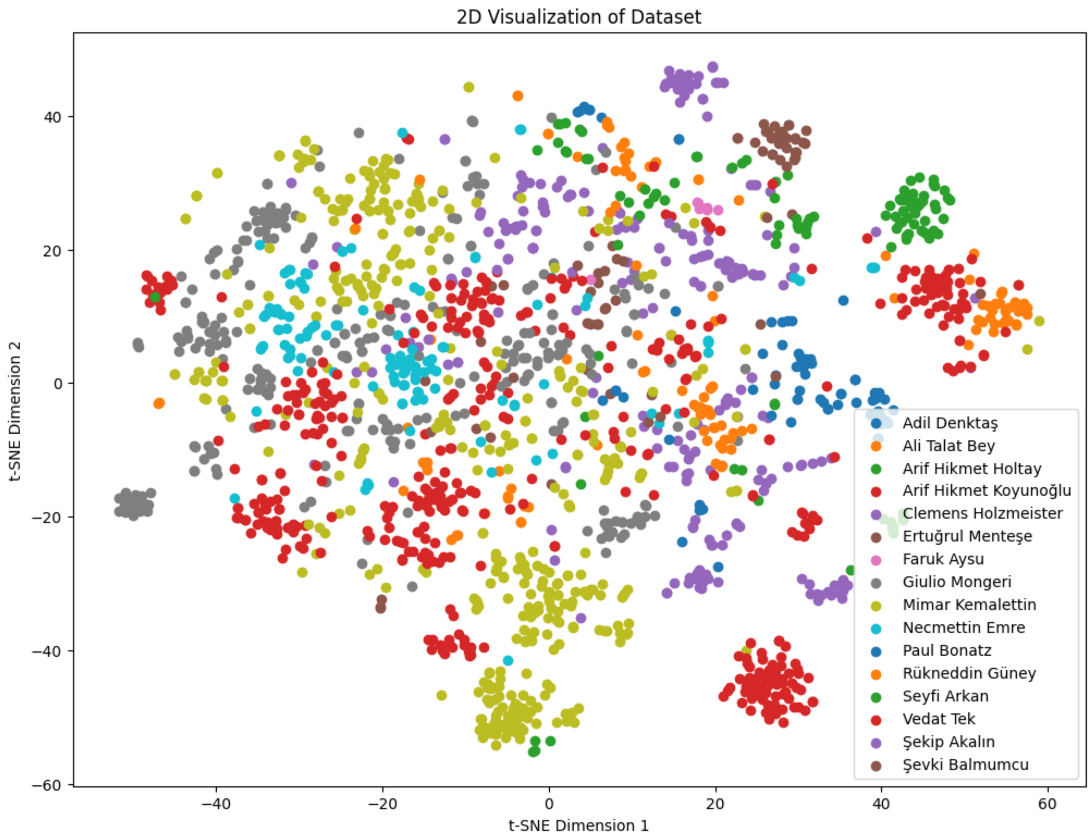


Figure 5.7: Projected deep features of Architects-All dataset.

# 6. CONCLUSION AND FUTURE WORK

In conclusion, this study presents a comprehensive exploration of architectural style classification from various neural network architecture perspectives. The development of a specifically crafted dataset for model training addresses the lack of appropriate datasets in this field. A careful analysis of four major Convolutional Neural Network (CNN) architectures – baseline, VGG, ResNet, and EfficientNet – reveals that EfficientNet is the most effective network architecture. Subsequently, with hyperparameter tuning and model training, accuracy rates of 84.65% for the Architects-Top dataset and 74.08% for the Architects-All dataset are achieved. Utilizing the last two models as feature extraction tools and applying t-SNE dimensionality reduction for visualization provides insights into similarities and differences among architects.

When evaluating future research directions, several significant areas emerge. The integration of additional architectural datasets has the potential to enhance the model's generalization abilities and foster a deeper understanding of various architectural styles. Furthermore, exploring more advanced neural network architectures and ensemble methods presents an opportunity to further improve classification accuracy. Additionally, the integration of trained models into practical applications could effectively assist architects in real-world scenarios during the design process and inspire the emergence of new designs through the use of generative artificial intelligence based on the obtained architectural features.

# BIBLIOGRAPHY


[1] D. Hasol, "Cumhuriyet dönemi mimarlığından bir panaroma," *İTÜ Sempozyum kitabı: Osmanlı Kültürel Mirası ve Mimarlıkta Süreklilik, Turkey*, 1999.

[2] M. Sözen, "Birinci ulusal mimarlık dönemi ve mimarları," *Mimaride Türk Millı Üslubu Semineri (11-12 Haziran)*, pp. 35–39, 1984.

[3] Ş. Çıkış, "Birinci ulusal mimarlık dönemi izmir konutu: Yerellik ve melezlik," *METU Journal of the Faculty of Architecture*, 2011.

[4] Z. Xu, D. Tao, Y. Zhang, J. Wu, and A. C. Tsoi, "Architectural style classification using multinomial latent logistic regression," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, Springer, 2014, pp. 600–615.

[5] S. Lee, N. Maisonneuve, D. Crandall, A. A. Efros, and J. Sivic, "Linking past to present: Discovering style in two centuries of architecture," in *IEEE International Conference on Computational Photography*, 2015.

[6] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. Efros, "What makes paris look like paris?" *ACM Transactions on Graphics*, vol. 31, no. 4, 2012.

[7] C. Pesto, *Classifying us houses by architectural style using convolutional neural networks*, 2017.

[8] Y. Yoshimura, B. Cai, Z. Wang, and C. Ratti, "Deep learning architect: Classification for architectural design through the eye of artificial intelligence," *Computational Urban Planning and Management for Smart Cities 16*, pp. 249–265, 2019.

[9] B. Wang, S. Zhang, J. Zhang, and Z. Cai, "Architectural style classification based on cnn and channel–spatial attention," *Signal, Image and Video Processing*, vol. 17, no. 1, pp. 99–107, 2023.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[12] M. Tan and Q. V. Le, *Efficientnet: Rethinking model scaling for convolutional neural networks*, 2020. arXiv: `1905.11946 [cs.LG]`.

[13] F. Zhuang, Z. Qi, K. Duan, *et al.*, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[14] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *Journal of Machine Learning Research*, vol. 18, no. 185, pp. 1–52, 2018.

[15] Dumitrux, *Architectural styles*, Aug. 2020. [Online]. Available: `https://www.kaggle.com/datasets/dumitrux/architectural-styles-dataset`.

# CV


## Emirkan Burak Yılmaz

Istanbul, Kartal / emirkan.b.yilmaz@gmail.com / +90 507 848 4803

LinkedIn / GitHub


## EDUCATION

**Gebze Technical University**, Kocaeli, Gebze

*Bachelor's of Engineering, Computer Engineering* *09/2019 - present*

Cumulative GPA: 3.34

Relevant Course Work: Object-Oriented Programming, Data Structures and Algorithms, Computer Organization, Operating Systems, System Programming, Natural Language Processing, Augmented Reality

## WORK EXPERIENCE

**Maastricht University Institute of Data Science**, Maastricht

*Data Science Intern (Erasmus Internship)* *06/2023 - 09/2023*

- Contributed to the development of a medical context dataset by extracting relevant information from drug labels in SPL format using the DailyMed database.
- Trained Named Entity Recognition (NER) models on a dataset with 21 tags, achieving an F1 score of 61. Conducted a comprehensive performance analysis by comparing these models with advanced large language models.

**PhiTech Bioinformatics**, Kocaeli, Gebze

*Web Development Intern* *06/2022 - 09/2022*

- Contributed to the collaborative development of a full-stack web application, optimizing backend logic, enhancing overall performance, and ensuring data integrity using technologies such as Spring Boot, Angular Material, and MySQL.

**Gebze Technical University**, Kocaeli, Gebze

*Tutor* *03/2022 - 06/2022*

- Assisted students with C language assignments in the Introduction to Programming course, adapting teaching methods to individual learning styles and collaborating with instructors.

## SELF PROJECTS

**MIPS32**
Designed a 32-bit single-cycle MIPS processor using Verilog HDL.
**GTUCAD**
Developed a 2D Computer-aided design (CAD) library in C language to facilitate the creation and manipulation of CAD content.

## SKILLS & INTERESTS

**Languages:** English, Turkish
**Programming Languages:** C, C++, Java, Python, JavaScript, Verilog HDL
**Hobbies:** Calisthenics, Reading, Exploring New Cities

# APPENDICES

## Links

Check out the trailer video on YouTube.
Explore the source code in our GitHub Repository.
Find the curated dataset on Hugging Face.
Access the trained models on Hugging Face.