Journal of Architectural Sciences and Applications
JASA 2023, 8 (2), 811-828
Research article
e-ISSN: 2548-0170
https://dergipark.org.tr/en/pub/mbud

Predicting Various Architectural Styles Using Computer Vision Methods

Meryem ÖZTÜRKOĞLU [1]*

ORCID 1: 0009-0004-8420-6096

[1] *Yıldız Technical University, Institute of Sciences Department of Architecture, 34349, Beşiktaş, Istanbul-Türkiye*

*** e-mail:** *meryemozturkoglu@gmail.com*

Abstract

Computer Vision (CV), subfield of artificial intelligence (AI), enables computers to process visual data and recognize objects. CV is widely used in, automotive, food industry and diseases diagnosis. AI achieves this by algorithms. One of the important algorithms based on object detection is YOLO (You Only Look Once), provides more accurate results with high processing speed. The aim of this study is to perform an object detection-based CV project, to determine the structures in given video belong to one of the architectural styles: Gothic, Baroque, Palladian, or Art Nouveau. The study consists of data set creation, data labeling, model creation and model training. Roboflow was used as the data labeling platform and YOLOv8 was used for model building and training phases. At the end of the process, the fact that the model predicts architectural styles with high accuracy in a short time revealed that the model is a successful real-time object detection algorithm, and it was emphasized that CV can be used in the field of architecture and can contribute to other fields related to architecture.

Keywords: *Computer vision, object detection, YOLO, architectural style.*

Computer Vision Metodlarıyla Çeşitli Mimari Üslupların Tahmin Edilmesi

Öz

Yapay zeka (AI) alanının alt dalı olan Computer Vision (bilgisayar görüşü, CV), bilgisayarların görsel verileri işleyerek nesneleri tanıyabilmesine olanak sağlar. CV, otomotiv, gıda endüstrisi, hastalıkların teşhisi gibi alanlarda yaygın kullanılmaktadır. AI bunu yaparken, algoritmaları kullanmaktadır. Nesne algılamaya dayalı algoritmaların en önemlilerinden biri yüksek veri işleme hızıyla daha net sonuçlar veren YOLO (You Only Look Once) dur. Bu çalışmanın amacı, temel alınan videodaki öne çıkan yapıların gotik, barok, palladyen, art nouveau mimari üsluplarından hangisine ait olduğunu belirlemeye yönelik nesne algılama tabanlı CV projesi gerçekleştirmektir. Çalışma veri seti oluşturma, veri etiketleme, model oluşturma ve modelin eğitimi aşamalarından oluşmaktadır. Veri etiketleme platformu olarak Roboflow, model oluşturma ve eğitim aşamaları için YOLOv8 kullanılmıştır. Süreç sonunda modelin mimari üslupları yüksek doğruluk payı ile kısa zamanda tahmin etmesi modelin başarılı gerçek zamanlı bir nesne algılama algoritması olduğunu ortaya koymuş, CV'ın mimarlık alanında da kullanılabileceği ve mimarlık ile ilgili diğer alanlara da katkı sunabileceği vurgulanmıştır.

Anahtar kelimeler: *Computer vision, nesne algılama, YOLO, mimari üslup.*

Citation: Öztürkoğlu, M. (2023). Predicting various architectural styles using computer vision methods. *Journal of Architectural Sciences and Applications*, 8 (2), 811-828.
DOI: https://doi.org/10.30785/mbud.1334044

Received: *28/07/2023 –* ***Accepted:*** *25/11/2023*

1. Introduction

Computer Vision (CV), a subfield of artificial intelligence, enables computers to process visual data and recognize objects. These data can range from still images, videos captured using traditional methods or more complex videos recorded with multiple cameras to multidimensional data from any medical scanner (Szeliski, 2010). The field of CV research dates to the 1960s when early methods for image recognition and classification were relatively simple. However, the advancement of deep learning techniques in recent years has led to significant progress in the field of CV. Today, CV is extensively used in various domains such as autonomous vehicles in the automotive industry, diagnosis of numerous diseases in the medical field, and determining the quality standards and existing quality of a product in the food industry, providing a significant advantage through the integration of artificial intelligence into daily life. CV encompasses a range of core tasks, including content recognition, video analysis, content-aware image editing, and scene reconstruction (Hosni, 2022). Among these fundamental tasks, "Object Detection" is considered a primary task within content recognition.

Object detection is a crucial CV task used to detect instances of visual objects in digital images such as photographs or video frames belonging to specific classes (e.g., humans, animals, cars, or buildings) (Boesch, 2023a). This modern approach of artificial intelligence develops computational models that provide essential information required by CV applications. It counts objects, processes their precise locations, and accurately labels the visual data being worked on, utilizing the same data. Object detection can be achieved using traditional image processing techniques or modern deep learning networks (convolutional neural networks - CNN). While traditional approaches require defining the features of the processed data and employing techniques like Support Vector Machines (SVM) for classification, deep learning approaches based on neural networks can perform end-to-end object detection without explicitly defining these features, following a one- or two-stage object detection algorithm (Contributors to Wikimedia projects, 2008). One of the most significant single-stage object detection algorithms is YOLO (You Only Look Once).

YOLO (You Only Look Once), developed by Joseph Redmon in 2016, stands out in this field due to its ability to detect objects in an image in a single pass, which results in faster performance compared to other object detection algorithms. Different versions of YOLO (e.g., YOLOv2, YOLOv3...Yolov8) have been developed, and ongoing research continues to improve these versions. YOLO has been utilized in various studies based on object detection. Kasper-Eulaers et al., (2021) used YOLOv5 to detect heavy-duty vehicles waiting in a parking area during the winter season. In another study, YOLO was used for crack detection in suspension parts (Özel, Baysal, & Şahin, 2021). Kristo, Ivasic-Kos, & Pobar (2020) conducted a study using YOLOv3 to detect humans in thermal camera images under challenging weather conditions. One of the areas where YOLO is applied is the field of architecture.

Architecture, in its simplest definition, is the art and science of designing and constructing buildings and other physical structures. Another definition characterizes it as the art and science of designing and constructing structures and the physical environment in appropriate proportions (Contributors to Wikimedia projects, 2001b). Vitruvius describes architecture as a concept that should be based on functionality, durability, and aesthetics (Vitruvius, 1999). Architectural style, on the other hand, refers to a specific design style that emerges from the combination of characteristic features of an architectural work, unique to a particular time and place. Historically, architectural styles reflect the aesthetic and cultural understanding of specific periods. For example, styles such as Gothic, Baroque, Palladian, and Art Nouveau provide distinct differences and insights into the architectural features of past eras.

1. Gothic Architecture: Originating in France in the 12th century, gothic architecture continued until the end of the 16th century and spread to various countries in Europe. Its aim was to build structures that symbolized the greatness of God and the helplessness of man in the face of God. The main gothic architectural elements are rib vaults, pointed arches, rose windows and flying buttresses. Two of the most important examples are Notre Dame Cathedral in Paris and Milan Cathedral in Milan.

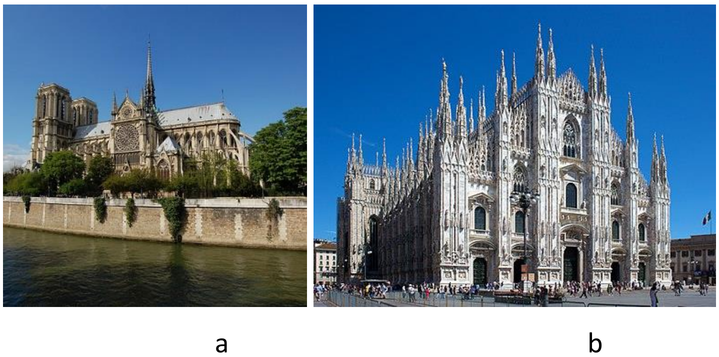


Figure 1. Gothic Architecture examples: **a)** Notre-Dame de Paris (Contributors to Wikimedia projects, 2002a) **b)** Milan Cathedral (Contributors to Wikimedia projects, 2003)

2. Baroque Architecture: Emerging in the 17th century in and around Rome, baroque architecture was mostly used in palace buildings as an effect of the desire of the royalty to show its power. The most important elements of baroque architecture, which has the understanding of using nature to give it a new form, are fountain pools, rich use of ornament and color, magnificent sculptures, frescoes on gods and mythology, and vase-shaped balusters. The most important examples include the Trevi Fountain in Rome and the Palace of Versailles in Paris.

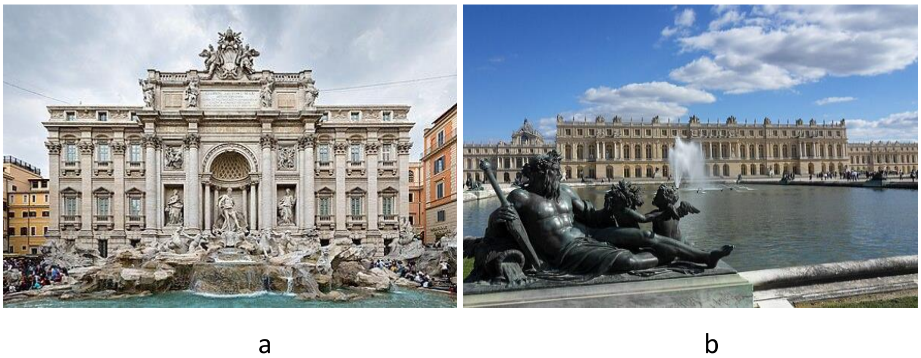


Figure 2. Baroque Architecture examples: **a)** Trevi Fountain (Contributors to Wikimedia projects, 2003a) **b)**Palace of Versailles (Contributors to Wikimedia projects, 2002)

3. Palladian Architecture: Palladianism is an architectural style named after the Venetian architect Andrea Palladio (1508-1580) and modeled after his designs. This movement emerged as a result of the evolution of Palladio's works and gained popularity in Europe. The works of Andrea Palladio are characterized by a pronounced symmetry and perspective, and often bear traces of classical temple architecture inspired by Ancient Greece and Ancient Rome. These elements, combined with Palladio's original concepts, shaped the Palladian architectural style. An important example of this architectural style is the Palladian Villas in Veneto, Italy.

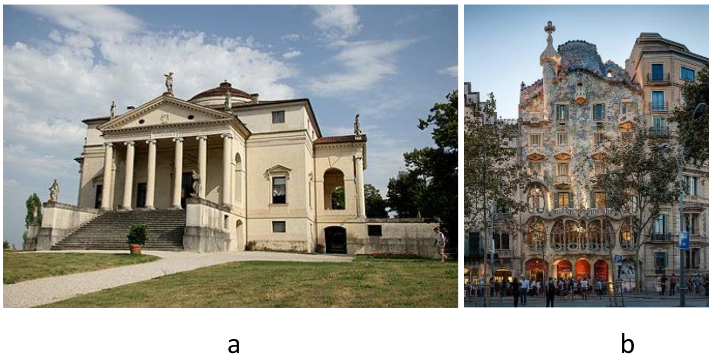


Figure 3. a) Palladian Architecture example: Palladian Villas (Contributors to Wikimedia projects, 2006) **b)** Art-Nouveau Architecture example: Casa Batllo (Contributors to Wikimedia projects, 2001)

4. Art-Nouveau Architecture: Art-Nouveau is an art movement pioneered by the famous Austrian painter Gustav Klimt and developed in Europe in the 19th century as a reaction to the Industrial Revolution. This movement includes elements such as the processing of iron for decorative purposes in response to the technological conditions of the age. Art-Nouveau is characterized by stylized, flattened, curved, asymmetrical and curved shapes. Rhythmic motifs, animals and plants, female figures, flying hair and feathers, flowers and vine shoots are important elements of this movement. These characteristics influenced the artists to design their works in accordance with the aesthetic understanding of the period. One of the most important examples is Antoni Gaudi's Casa Batllo in Barcelona.

These architectural styles exhibit similar repetitive patterns manifesting at different scales (Yıldız, Ertosun Yıldız & Beyhan, 2023). Determining the architectural style of a structure involves distinguishing fundamental architectural elements such as forms, materials, and details that may exhibit variations or similarities.

This study aims to demonstrate that CV can also be used in the field of architecture by realizing a CV project based on object detection using YOLOv8 to determine whether the prominent structures in a given video belong to one of four architectural styles (Gothic, Baroque, Palladian or Art Nouveau).

The application of these technologies in architecture is important for the analysis and recognition of architectural styles. Artificial intelligence algorithms enable designers, engineers and architects to perform fast and precise analysis by accurately identifying the characteristics of structures. As a matter of fact, the number of studies focusing on architectural style identification using artificial intelligence techniques is limited in the literature (Xu, Tao, Zhang, Wu, & Tsoi, 2014). Therefore, considering the method and approach based on the basic program used in this study and the original results targeted to be obtained, it is thought that it will make a significant contribution to the literature by differentiating from the others.

2. Material and Method

In this study, we utilized YOLOv8, a state-of-the-art single-stage computer vision (CV) algorithm specifically designed for object detection tasks. YOLOv8 has proven to be highly effective in accurately identifying and localizing objects in complex visual scenes.

To train the model, we created a comprehensive dataset consisting of images and videos showcasing prominent structures with various architectural styles, including Gothic, Baroque, Palladian, and Art Nouveau. Each image and video frame were carefully annotated to provide ground truth labels for the architectural style of the structures.

The training process involved feeding the dataset into the YOLOv8 model and optimizing its parameters through a series of iterative epochs. During training, the model learned to extract meaningful features from the input data and associate them with the corresponding architectural styles. The training was performed on a powerful GPU-enabled system, allowing for efficient computation and accelerated convergence.

To evaluate the performance of our approach, we conducted a thorough validation process. We used a separate validation dataset that included diverse images and videos representing different architectural styles. The model was tested on this dataset, and various performance metrics were calculated, including precision (P), recall (R), mean average precision at 50% overlap (mAP50), and mean average precision from 50% to 95% overlap (mAP50-95). These metrics provide insights into the accuracy and robustness of the model's predictions.

Furthermore, we employed a confusion matrix analysis to assess the model's ability to correctly classify structures into their respective architectural styles. The confusion matrix provided a comprehensive overview of the model's prediction accuracy for each architectural style, allowing us to evaluate its performance in a more detailed and specific manner.

In addition to evaluating the model's accuracy, we also analyzed its inference speed. The object detection process was applied to each frame of the video using the trained model, and the time taken for processing each frame was measured. This analysis provided valuable insights into the model's efficiency and its suitability for real-time applications.

Overall, our methodology combined the power of YOLOv8 with a carefully curated dataset, extensive training, rigorous validation, and performance evaluation to achieve accurate architectural style determination. The comprehensive approach employed in this study ensures the reliability and robustness of our results.

2.1. Computer Vision (CV) and YOLO (You Only Look Once) Algorithm

According to Trucco & Verri (1998), CV is the calculation of three-dimensional world properties from one or more images. Research conducted by Chang Shu defines CV as a field that involves extracting useful information from digital image contents and employing computer models to simulate the way living beings perceive through their eyes, thereby interpreting and understanding images (Su, 2008). The aim in this field is to mimic the process of visual perception and comprehension performed by human eyes using computer models by extracting information from the contents of images and videos. The objectives of CV tasks include enabling computer systems to automatically perceive, recognize, and understand the visual world using computational methods to simulate human vision (Boesch, 2023).

CV as a field utilizes the latest technologies, such as deep learning methods, which are a subfield of machine learning, to train algorithms. Currently, many of the methods employed in CV are based on layered neural networks called convolutional neural networks (CNN). Convolutional operations, which have been used in mathematics, physics, and engineering applications to simplify complex calculations, have yielded the best results in the field of computer vision (CV). When processing image data, a CNN network is applied to "look at" the computer data. CNN breaks down labeled pixels to allow the deep learning model to understand the images. Artificial intelligence models make predictions about the images using these labels. The accuracy of the obtained predictions is repeatedly checked until the desired outcome is achieved. Artificial neural network models, first developed in 1943 (Elmas, 2018) by Warren McCulloch, a neurologist, and Walter Pitts, a mathematician, indicate their potential significance, particularly in enhancing the quality of daily life for robots in the near future (Efe & Kaynak, 1999).

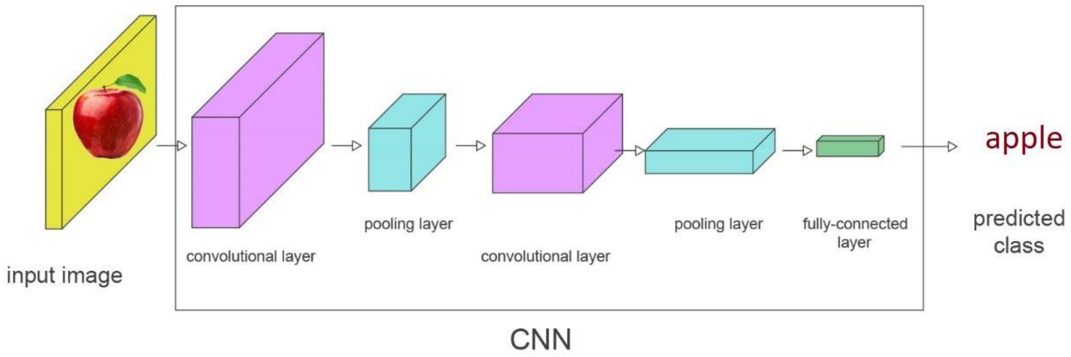


Figure 4. CNN neural network architecture diagram

1. Convolutional Layer: Extracts relevant features from input images by applying learnable filters and generating feature maps. Enables the model to automatically learn and differentiate architectural features associated with different styles.
2. Pooling Layer: Reduces spatial dimensions of feature maps while preserving important information. Enhances computational efficiency and robustness to transformations. Retains prominent features and suppresses noise.

3. Fully Connected Layer: Connects neurons from previous layers to classify architectural styles. Maps high-level features to style classes, determining the architectural style of structures in the video.

These layers collectively form a neural network architecture for automatic detection and classification of architectural styles, improving accuracy and enabling comprehensive analysis of the prominent structures.

Computer Vision (CV) encompasses various tasks and techniques for understanding and analyzing visual content. In this section, we will summarize key CV tasks and their relevance to our study.

1. Content Recognition: Content recognition involves recognizing and categorizing visual content. Image classification is a core task in CV, where models are trained to assign labels to images based on their features. Convolutional Neural Networks (CNNs) are commonly used for image classification, leveraging large datasets and backpropagation for accurate recognition.

   1.1. Object Detection: Object detection identifies and localizes multiple objects in images, providing both class labels and bounding box coordinates. This task is crucial for our study as it determines the architectural style of prominent structures. YOLO and Faster R-CNN are popular algorithms for real-time object detection.

   1.2. Object Localization: Object localization focuses on precisely locating a single object within an image. It provides the bounding box coordinates enclosing the object of interest, enabling precise analysis of architectural features.

   1.3. Object and Instance Segmentation: Object segmentation partitions an image into regions corresponding to different objects, assigning labels to pixels or regions. Instance segmentation goes further, distinguishing individual instances of the same object class. While not directly used in our study, these tasks are important for spatial understanding in CV applications.

   1.4. Pose Estimation: Pose estimation estimates the spatial orientation of an object in an image or video. It determines the object's position, scale, and orientation relative to a reference system. Pose estimation can aid in understanding the spatial arrangement of architectural structures.

2. Video Analysis: CV techniques can be applied to analyze videos, including object tracking, action recognition, and motion prediction.

   2.1. Object Tracking: Object tracking follows objects across consecutive video frames, enabling the study of their movement and behavior over time.

   2.2. Action Recognition: Action recognition identifies and classifies human actions in video sequences. It helps analyze human interactions with architectural structures.

   2.3. Motion Prediction: Motion prediction forecasts future positions and movements of objects based on their past trajectories, assisting in understanding and predicting the behavior of structures in a video.

3. Content-Aware Image Editing: CV techniques can be used for automatic understanding and modification of specific image aspects while preserving overall quality. Content-aware image editing is valuable for architectural analysis and visualization, allowing modifications while maintaining architectural integrity.

4. Scene Reconstruction: Scene reconstruction creates a 3D model of a real-world scene from 2D images or videos. CV algorithms enable accurate representation of architectural scenes, supporting immersive visualization and precise measurements in architectural analysis and design.

These various CV tasks and applications provide the foundation for our study, enabling us to leverage object detection and architectural style determination in the context of video analysis. By utilizing these techniques, we can gain valuable insights into the architectural characteristics and styles exhibited by prominent structures within a video.

In computer vision, particularly for real-time object detection, both multi-stage and single-stage algorithm groups based on CNN are used. Multi-stage algorithms are more accurate but slower than single-stage algorithms. Multi-stage algorithms, exemplified by R-CNN models such as Mask-RCNN, Fast RCNN, and Faster RCNN, excel in achieving higher levels of precision but operate at a slower pace. Conversely, single-stage algorithms like SSD, RetinaNet, YOLOv3, YOLOv4, YOLOR, YOLOv5, YOLOv7, and YOLOv8 prioritize real-time processing and computational efficiency overachieving the utmost accuracy (Boesch, 2023).

YOLO (You Only Look Once), developed by Joseph Redmon and Ali Farhadi from the University of Washington, was introduced in 2015 and quickly gained popularity due to its high speed and accuracy in object detection and image classification (Jocher, Waxmann, & Chaurasia, 2023). Unlike R-CNN algorithms, YOLO does not make any prior position predictions. Instead, YOLO processes the entire image to extract the locations of objects and their corresponding classes in a single pass. This approach is based on treating object detection as a single regression problem. The YOLO algorithm surrounds the detected objects on the images with bounding boxes (bb). To achieve this, the input image is first divided into grids of size s x s (such as 5x5, 7x7, 9x9). Each grid considers whether there is an object within its own area and if the object is considered to have its center point within its own area. The grid that determines the object's center point then predicts its x-coordinate, y-coordinate, width, and height. Additionally, the boxes include a confidence score indicating whether the object belongs to the defined class. The YOLO network detects and classifies objects using the confidence scores of these bounding boxes.

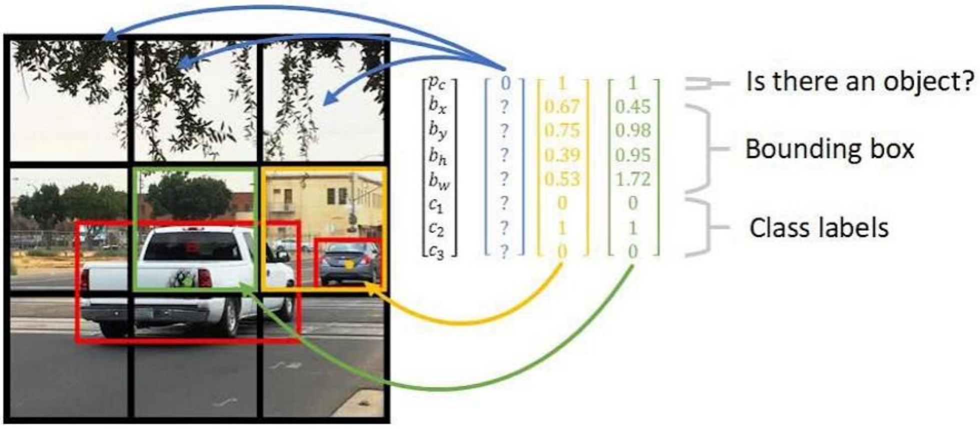


Figure 5. YOLO grid system and bounding boxes (Handuo, 2018)

The confidence score indicates how confident the model is about the presence of an object within the current grid. Confidence score formula is:

Confidence Score = Pr(object) x IoU

Pr(object): Probability of the object being present within the grid.

IoU: Overlap between the ground truth box and the predicted box.

When the model considers that there is an object within the grid, it checks how certain it is about whether the object is indeed that object and the coordinates of the surrounding box. Multiple grids may consider that the object is within their own area, resulting in unnecessary bounding boxes on the screen. To address this, the Non-max Suppression algorithm is primarily used in object detection, aiming to select the best bounding box from a series of overlapping bounding boxes. It considers only the ones with a confidence score above a certain threshold as the final output and draws them on the screen (Redmon, Divvala, Girshick, & Farhadi, 2016).

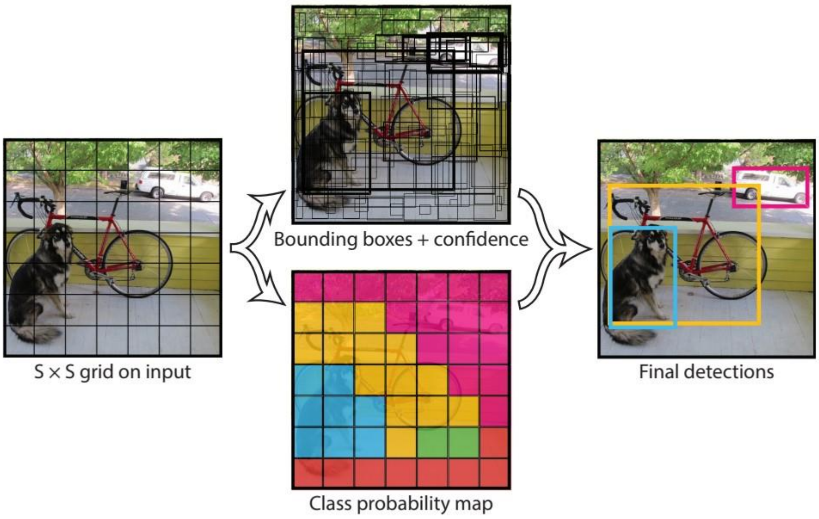


Figure 6. YOLO bounding box working system (Redmon et al., 2016)

Different versions of YOLO (Figure 7) have been introduced since its initial development until the present day, including YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOR, YOLOv5, and YOLOv8. YOLOv8, introduced in 2023, is the latest and most advanced version of the YOLO object detection and image segmentation model. Building upon the success of previous versions, this latest model offers new features and improvements in terms of performance, flexibility, and efficiency (Figure 8). YOLOv8 is designed with a focus on factors such as speed, size, and accuracy, making it highly effective for various artificial intelligence tasks. This versatility allows users to leverage the capabilities of YOLOv8 in different application domains. In this study, the YOLOv8 version was used.

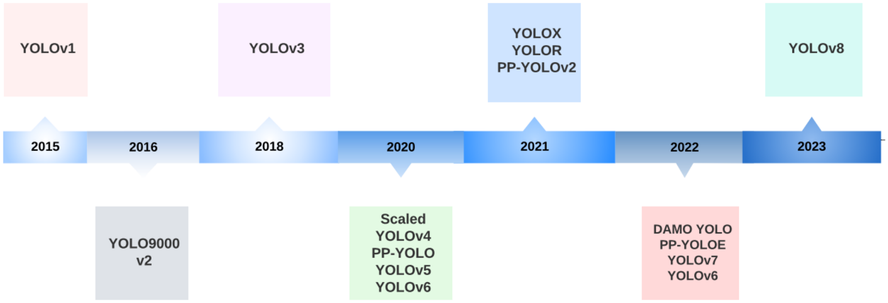


Figure 7. Timeline of YOLO versions (Terven & Cordova-Esparza 2023)

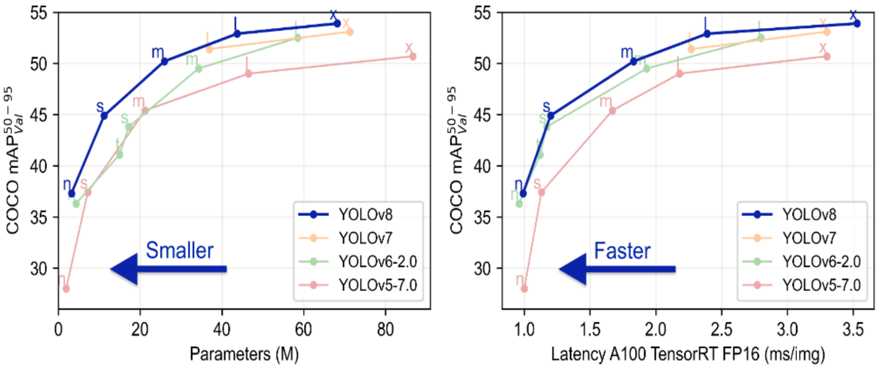


Figure 8. Performance comparison of YOLOv8 with other versions (Jocher & Waxmann, 2023)

The studies conducted to reach the conclusion based on the definitions mentioned above were carried out in three stages.

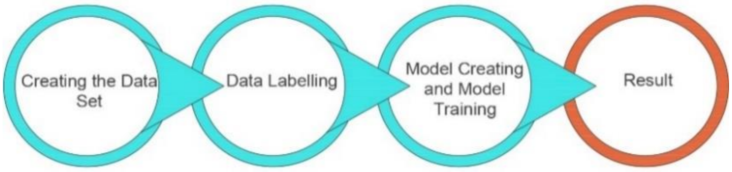


Figure 9. Stages of the study

2.2. Dataset Creation

The initial stage of this study involved the creation of the dataset, which progressed through four selected architectural styles: Gothic, Baroque, Palladian, and Art Nouveau. (In subsequent works, it is planned to expand the number of these styles.) Photographs corresponding to each architectural style were collected from free stock photo websites and the online community platform Kaggle (Wwymak, n.d), which consists of data scientists and machine learning practitioners. The dataset, comprising 1434 images (Figure 10), was transferred to the Roboflow platform for data labeling (Öztürkoğlu, 2023). Roboflow is a computer vision platform that enables users to create CV models faster and more accurately by providing improved data collection, preprocessing, and model training techniques (Roboflow, 2020).

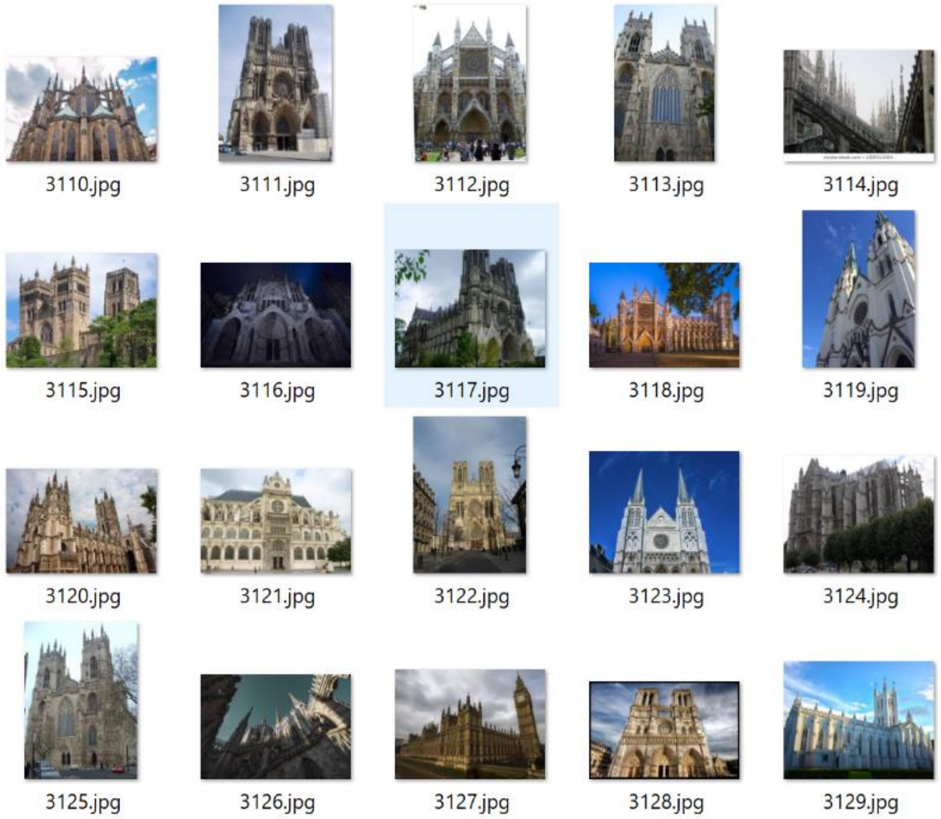


Figure 10. Photos from the dataset

2.3. Data Labelling

CV models cannot comprehend raw data in their original form. Therefore, the categorization and labeling of data provide meaning to the CV models. Labeled data is used as the training set and helps models produce accurate results. This process involves annotating and categorizing meaningful parts of the data.

There are various data labeling platforms in the literature (Sager, Janiesch, & Zschech, 2021). Some of them include LabelMe, LabelImg, VIA, and Image Tagger. In this study, the Roboflow platform was used for data labeling. Roboflow allows users to upload custom datasets, add descriptions, change image orientations, resize images, adjust image contrast, and perform data augmentation for model training.

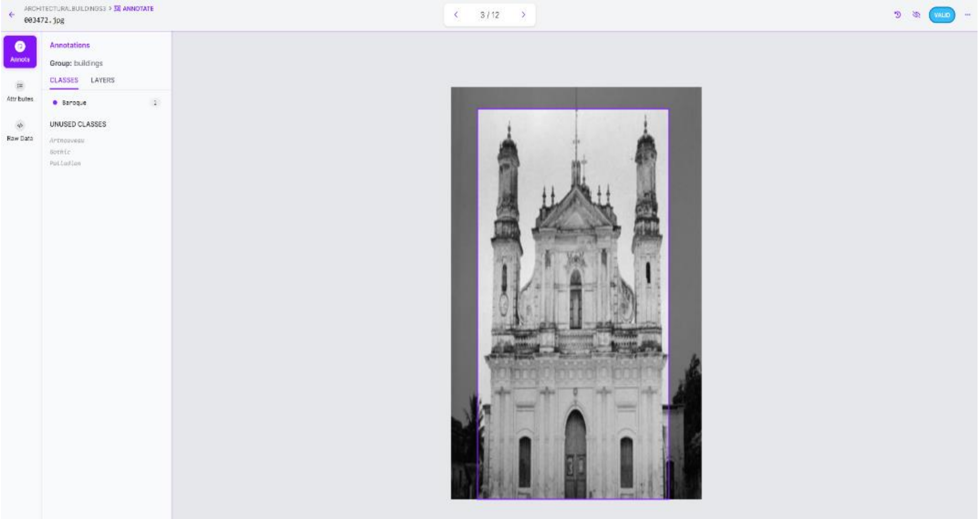


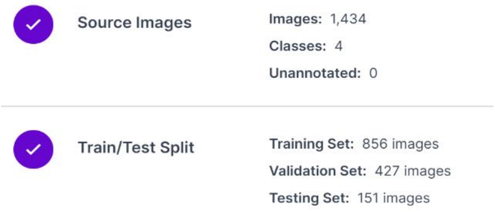
Figure 11. Data labeling on the Roboflow platform

2.4. Model Creation and Model Training

The process of data collection, adding annotations and retraining is referred to as "active learning", training a model from a checkpoint of a previous model is known as "transfer learning" (Williams, 2021). In this study, the active learning method was employed.

2.4.1. Model creation

To achieve desirable accuracy and consistency performance, the dataset was divided into training, validation, and test parts. The training process involves optimizing the model's parameters, enabling accurate prediction of object classes and their locations in an image. This separation was performed using the Roboflow platform. Accordingly, the dataset consisting of 1434 images with four classes (Gothic, Baroque, Palladian, Art Nouveau) was split into 60% for training, 30% for validation, and 10% for testing.



Source Images

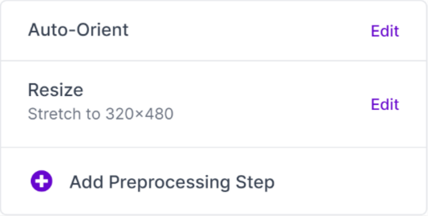
Images: 1,434
Classes: 4
Unannotated: 0

Train/Test Split

Training Set: 856 images
Validation Set: 427 images
Testing Set: 151 images

Figure 12. Train/test split on the Roboflow platform

The validation process is used after model training for validation purposes. It is employed to measure the model's accuracy and generalization performance. The test process measures the model's ability to predict previously unseen data. After dividing the dataset into train/validation/test, the images underwent a preprocessing step. Image preprocessing includes formatting steps such as resizing, color correction, and orientation adjustment. These steps are employed to reduce training time and increase model prediction speed. Resizing large images, in particular, improves training time without compromising model performance.



Auto-Orient
Edit

Resize
Stretch to 320×480
Edit

Add Preprocessing Step

Figure 13. Preprocessing process on the Roboflow platform

In this process, auto-orient (automatic orientation) was applied as the first step, followed by resizing as the second step. Depending on how the pixels of an image are stored, the data determines how the image is displayed. However, some image viewers may display images in the wrong direction without using this data. Auto-orient prevents this misdirection. The second step, resizing, aims to reduce all the images in this study to the same size, thus improving model speed.

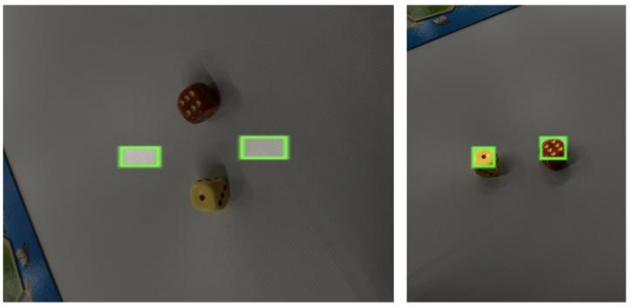
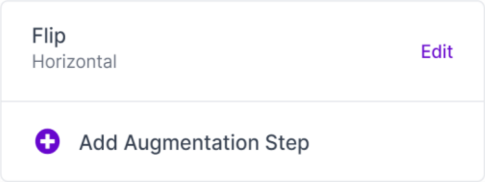


Figure 14. Left: Inconsistency between additional annotation and image orientation, Right: Corrected orientation with auto-orient (Dwyer, 2020)

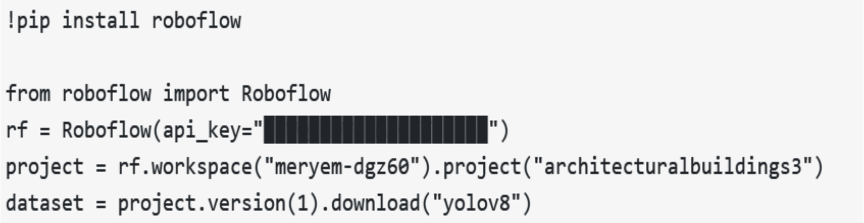
Image augmentation is a process used to increase the accuracy of the model by generating synthetic training data and improving the model's prediction performance. In this stage of model creation, the flip operation was applied, flipping the images horizontally.



Flip
Horizontal

Figure 15. Flip augmentation process

To increase the number of images in the dataset, the generating process was employed in this stage of model creation. As a result, the dataset consisting of 1434 images was transformed into a dataset consisting of 3146 images. In the final step, the model was exported in the YOLOv8 format, enabling its extraction from the platform (Figure 16).



```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="[redacted]")
project = rf.workspace("meryem-dgz60").project("architecturalbuildings3")
dataset = project.version(1).download("yolov8")
```

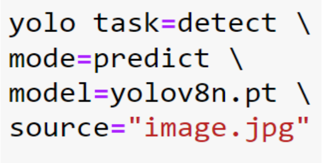
Figure16. Export process

By completing all the above steps, the model was created and prepared for training.

2.4.2. Model training and validation

The model training in this study was performed using the notebook (a workspace created by Python code) on Google Colab (Öztürkoğlu, 2023a), which provides free usage service. To utilize the full potential of YOLOv8, the ultralytics package was installed to meet the requirements.

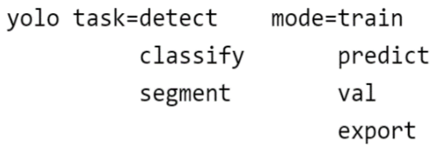
YOLOv8 provides both a comprehensive Command Line Interface (CLI) API and a Python SDK for training, validation, and prediction (Rath, 2023). In this study, the CLI was used. Figure 17 illustrates an example command line for running an object detection task using the CLI.



```
yolo task=detect \
mode=predict \
model=yolov8n.pt \
source="image.jpg"
```

Figure 17. YOLO CLI example usage

YOLO can perform three tasks: detection, classification, and segmentation. In this study, the "detect" task of YOLO was utilized. As for the mode, there are training (train), validation (val), prediction (predict), and export modes used to export a trained model.



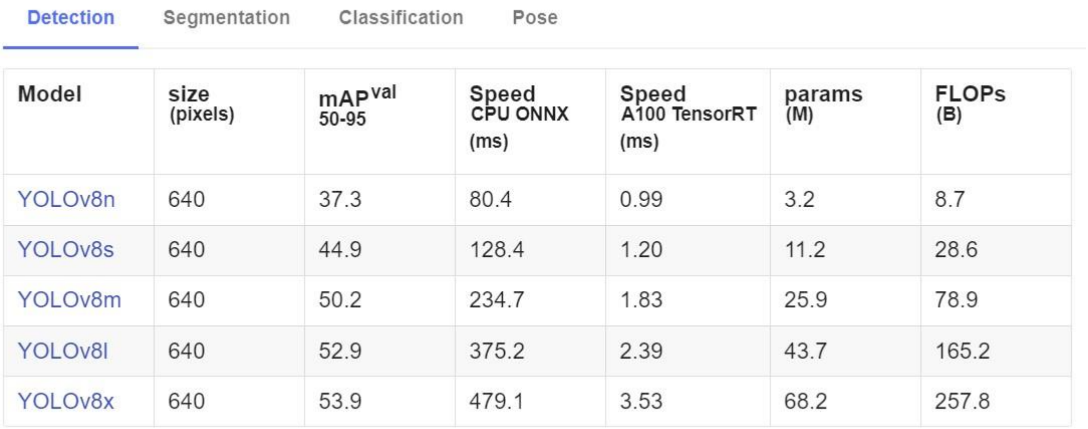
```
yolo task=detect    mode=train
          classify       predict
          segment        val
                         export
```

Figure 18. YOLO task and mode types

In this study, the command "mode = train" was used for model training, "mode = val" for model validation, and "mode = predict" for predicting the test data. Each category of YOLOv8 models has five models for detection, segmentation, and classification.

Upon examining the values in Table 1, it can be observed that YOLOv8x outperforms other versions in terms of model detection performance. YOLOv8 nano is the fastest and smallest, while YOLOv8 extra-large (YOLOv8x) is the most accurate but slowest version. In this study, the "yolov8s (small)" model was used.

Table 1. Object detection performance comparison of YOLOv8 nano, small, medium, large, xlarge models (Jocher & Waxmann, 2023)



Detection
Segmentation
Classification
Pose

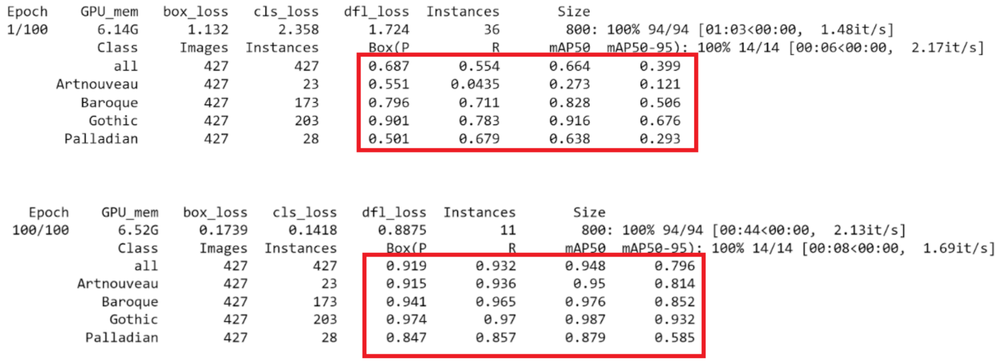
Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

After splitting the dataset into a training set, a validation set, and a test set with proportions of 60%, 30%, and 10%, respectively, the model was trained on a desktop computer equipped with a Windows 10 (64-bit) operating system, an AMD Ryzen 5 1600X Six-Core processor running at 3.60 GHz, and 16 GB RAM. The training process lasted approximately 1 hour and 40 minutes, consisting of 100 epochs. In the context of computer vision learning, an epoch refers to a single forward and backward pass of the training data through the neural network via an algorithm. When an epoch's worth of data is larger than what the computer can handle, it is divided into smaller parts called "batches." The number of batches required to complete an epoch is referred to as an "iteration" (Simplilearn, 2022).

3. Findings and Discussion

In this study, the aim was to determine the architectural style of a structure among Gothic, Baroque, Palladian, and Art Nouveau styles using YOLOv8, a one-stage algorithm based on object detection in computer vision (CV). The evaluation of the model's performance was conducted through measurements between the 1st and 100th epochs, using the training and validation sets, as well as the confusion matrix and loss functions. These evaluations were presented in Figures 19, 20, 21, 22, and 23, including numerical values and graphs.

Figure 19 illustrates the values of precision (P), recall (R), and mean average precision (mAP) between the 1st and 100th epochs. It can be observed that these values gradually increase. High P and R values indicate that the model performs well in correctly identifying all true positive detections and minimizing false negatives. Additionally, high mAP50 and mAP50-95 values indicate the model's effectiveness in detecting and localizing objects in different categories. Values close to 1 for P, R, and mAP indicate successful training processes on the training and validation datasets.

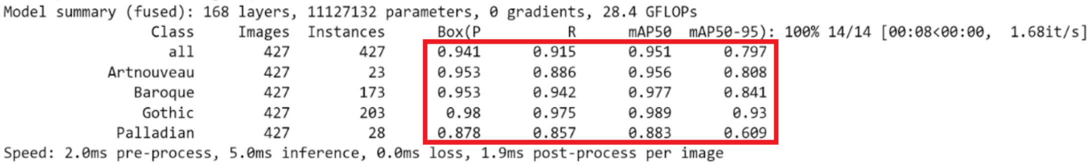


```
Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
1/100      6.14G      1.132      2.358      1.724         36        800: 100% 94/94 [01:03<00:00,  1.48it/s]
           Class     Images  Instances      Box(P          R       mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.17it/s]
             all        427        427      0.687      0.554      0.664      0.399
     Artnouveau        427         23      0.551     0.0435      0.273      0.121
        Baroque        427        173      0.796      0.711      0.828      0.506
         Gothic        427        203      0.901      0.783      0.916      0.676
      Palladian        427         28      0.501      0.679      0.638      0.293

   Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
 100/100      6.52G     0.1739     0.1418     0.8875         11        800: 100% 94/94 [00:44<00:00,  2.13it/s]
             Class     Images  Instances      Box(P          R       mAP50  mAP50-95): 100% 14/14 [00:08<00:00,  1.69it/s]
               all        427        427      0.919      0.932      0.948      0.796
        Artnouveau        427         23      0.915      0.936       0.95      0.814
           Baroque        427        173      0.941      0.965      0.976      0.852
            Gothic        427        203      0.974       0.97      0.987      0.932
         Palladian        427         28      0.847      0.857      0.879      0.585
```

Figure 19. Values between the 1st and 100th epochs

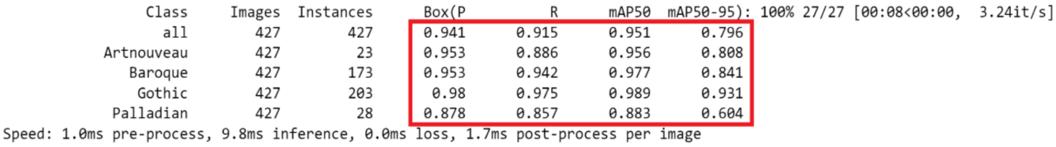
The mAP values, which measure how well the model accurately detects and classifies objects in different categories, reaching values close to 1 after the training (Figure 20), demonstrate the success of the training process.



```
Model summary (fused): 168 layers, 11127132 parameters, 0 gradients, 28.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:08<00:00,  1.68it/s]
                   all        427        427      0.941      0.915      0.951      0.797
            Artnouveau        427         23      0.953      0.886      0.956      0.808
               Baroque        427        173      0.953      0.942      0.977      0.841
                Gothic        427        203       0.98      0.975      0.989       0.93
             Palladian        427         28      0.878      0.857      0.883      0.609
Speed: 2.0ms pre-process, 5.0ms inference, 0.0ms loss, 1.9ms post-process per image
```

Figure 20. Training values

The validation values (Figure 21) were similar to the training values, indicating the success of the validation process.



```
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 27/27 [00:08<00:00,  3.24it/s]
                   all        427        427      0.941      0.915      0.951      0.796
            Artnouveau        427         23      0.953      0.886      0.956      0.808
               Baroque        427        173      0.953      0.942      0.977      0.841
                Gothic        427        203       0.98      0.975      0.989      0.931
             Palladian        427         28      0.878      0.857      0.883      0.604
Speed: 1.0ms pre-process, 9.8ms inference, 0.0ms loss, 1.7ms post-process per image
```

Figure 21. Validation values

According to the confusion matrix results in Figure 22, the model correctly predicts art nouveau structures with an accuracy of 0.87, baroque structures with an accuracy of 0.96, gothic structures with an accuracy of 0.98, and palladian structures with an accuracy of 0.89. However, it mistakenly predicts art nouveau structures as baroque with an accuracy of 0.09 and baroque structures as gothic with an accuracy of 0.01. The lower accuracy in predicting art nouveau structures compared to other styles can be attributed to the smaller number of labeled photos belonging to the art nouveau style. As the number of labeled photos increases, the model's accuracy also improves.

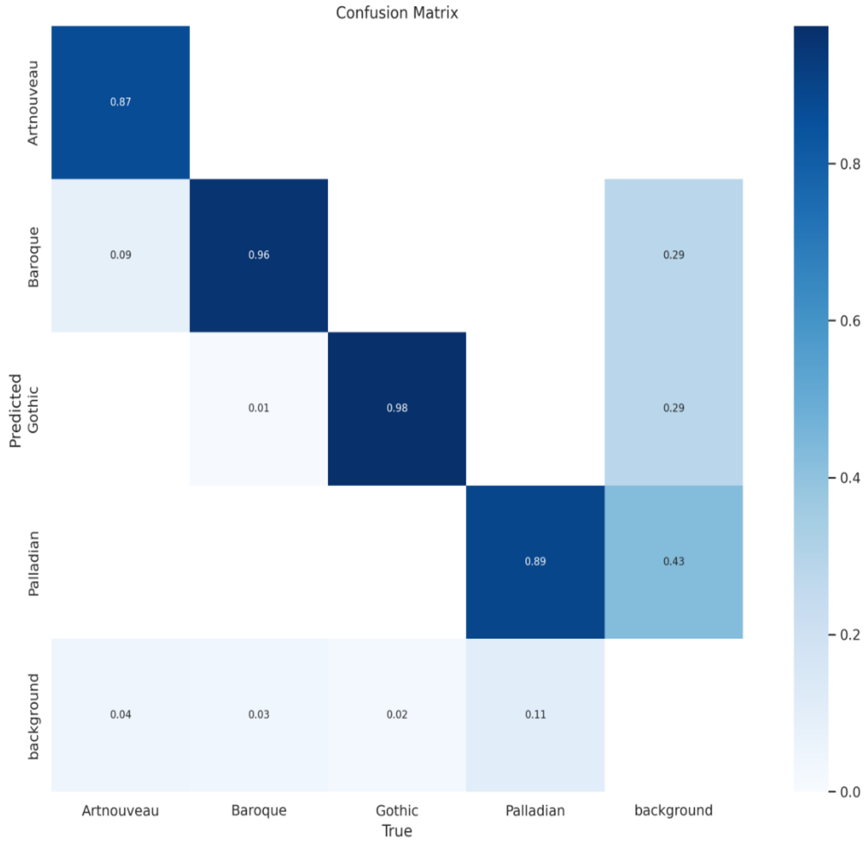


Figure 22. Confusion matrix

Figure 23 presents the results of the training and validation sets using graphs. These graphs show the measurements of three different loss types (box_loss, cls_loss, dfl_loss) as well as P, R, mAP50, and mAP50-95 values. Box_loss represents how well the algorithm can find the center of an object and how accurately the predicted bounding boxes (bb) cover an object. Cls_loss measures how well the algorithm predicts the class of an object. Dfl_loss is a measurement directly optimizing the distribution of bb boundaries. After 25 epochs, the model has significantly improved in terms of P, R, and mAP values, and it becomes stable after 50 epochs. This indicates that early stopping could provide almost similar results in 50% less time.

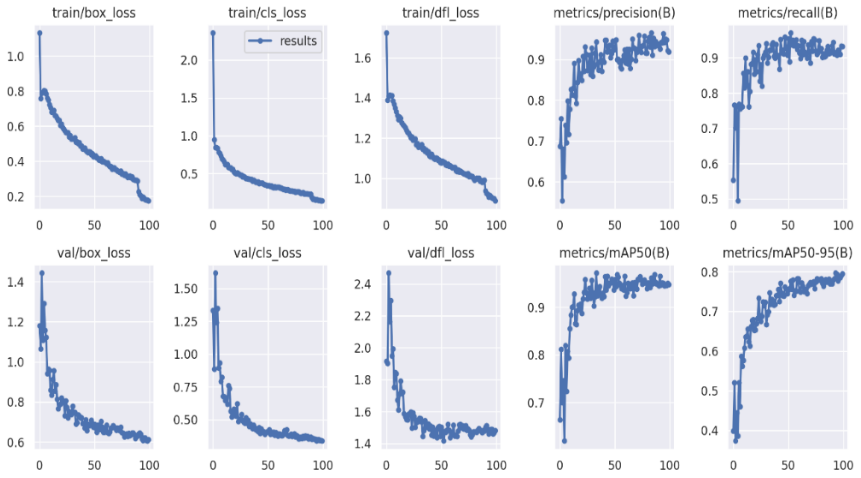


Figure 23. Graphs showing box_loss, cls_loss, dfl_loss, P, R, and mAP during the training period on the training and validation sets

After training the model, unseen photos (test set) and a video (Öztürkoğlu, 2023c) were shown to the model with a confidence threshold of 0.25 for inference. The algorithm was able to detect the architectural styles of structures with an accuracy of over 0.90 in most cases (Figure 24). The entire video, including the images in the figure and their accuracy rates, can be viewed on YouTube (Öztürkoğlu, 2023d).

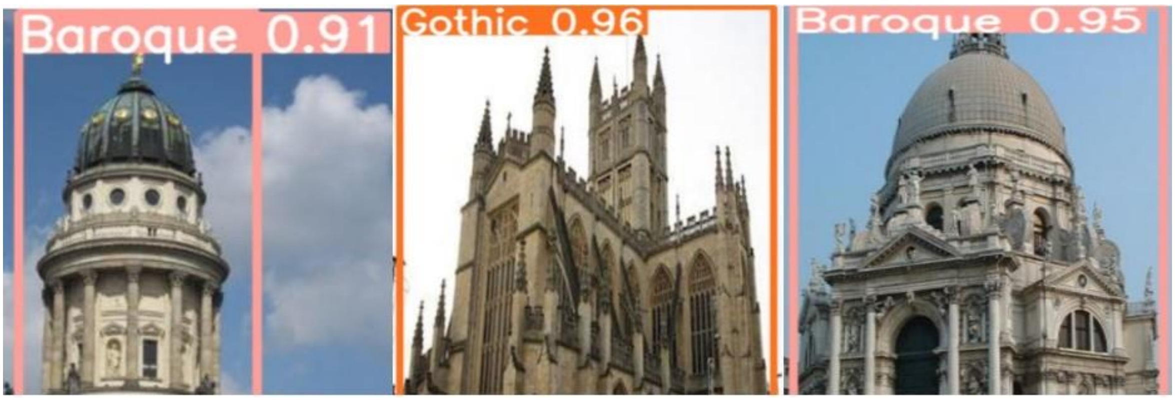
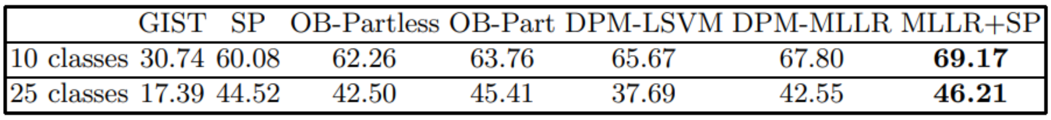


Figure 24. Predicted values of test photos

Furthermore, when observing the object detection time for each frame of the video, it took between 8-17 ms to process each frame. This value indicates that the detection of structures belonging to four different architectural styles with an average accuracy of 0.80-0.95 is almost real-time object detection.

Xu et al. (2014), conducted a study titled "Architectural Style Classification Using Multinomial Latent Logistic Regression", where they used a dataset consisting of photos of 25 different architectural styles to classify architectural styles using a Multinomial Latent Logistic Regression (MLLR) based model. In their study, they compared the prediction values of a 10-class model and a 25-class model, which they presented in Table 2. According to their findings, the prediction values of their model were below 0.70 for the 10-class model and below 0.50 for the 25-class model.

Table 2. Prediction results of Xu's study (Xu et al.2014)



	GIST	SP	OB-Partless	OB-Part	DPM-LSVM	DPM-MLLR	MLLR+SP
10 classes	30.74	60.08	62.26	63.76	65.67	67.80	**69.17**
25 classes	17.39	44.52	42.50	45.41	37.69	42.55	**46.21**

Architectural styles emerge and change over time, influenced by factors such as the geographical, demographic, and sociological characteristics of societies. In this sense, determining the architectural style of a structure paves the way for identifying the characteristics of the society to which it belongs. Reaching a conclusion about the architectural style of a structure requires the accumulated knowledge and expertise in that field by architects, designers, engineers, or artists. This study highlights the potential use of CV in intricate tasks like architectural style identification, emphasizing the contribution and efficiency that artificial intelligence algorithms can provide in this field. As a next step, this study can be extended by labeling and incorporating photos of structures belonging to other architectural styles into the model, allowing for more comprehensive utilization of the model in architecture, engineering, art, and related disciplines.

4. Conclusion and Suggestions

In this study, a CV project based on object detection using YOLOv8 was successfully conducted to determine the architectural style of prominent structures in a video, specifically focusing on the Gothic, Baroque, Palladian, and Art Nouveau styles. A comprehensive dataset was created, and the model was subjected to a rigorous training process consisting of 100 epochs. The results obtained from the study are highly promising and provide valuable insights into the capabilities of the approach employed:

Upon concluding the training process, our model achieved impressive performance metrics for all architectural styles. The precision (P), recall (R), mean Average Precision at IoU 0.50 (mAP50), and mean Average Precision at IoU 0.50-0.95 (mAP50-95) values were measured at 0.941, 0.915, 0.951, and 0.797, respectively. These exceptional ratios clearly demonstrate the effectiveness of our training procedure in capturing the distinctive features of different architectural styles.

Similarly, the validation process yielded remarkable results, with the model exhibiting P, R, mAP50, and mAP50-95 values of 0.941, 0.915, 0.951, and 0.796, respectively, for all architectural styles. These metrics further validate the robustness and generalizability of our model, highlighting its ability to accurately identify architectural styles in unseen data.

Moreover, our confusion matrix analysis revealed the model's high accuracy in predicting architectural styles. Specifically, the model demonstrated an outstanding accuracy of 0.98 for Gothic structures, 0.96 for Baroque structures, 0.89 for Palladian structures, and 0.87 for Art Nouveau structures. These findings provide strong evidence of the model's reliability and its ability to correctly classify diverse architectural styles.

In the video used for inference, our model exhibited an impressive average accuracy range of 0.80 to 0.95 when identifying structures belonging to the four different architectural styles. This demonstrates the model's capability to make accurate predictions consistently, even in the presence of challenging variations in lighting conditions, angles, and environmental factors.

Furthermore, the object detection process for each frame of the video demonstrated remarkable efficiency, with a processing time ranging from 8 to 17 milliseconds. This real-time performance, coupled with high prediction values, signifies that our model can effectively operate as a practical real-time object detection algorithm for architectural style recognition tasks.

In conclusion, this study showcases the significance of computer vision (CV) in the field of architecture and emphasizes its potential contribution to related domains. By accurately identifying and classifying architectural styles, our model provides a valuable tool for architectural analysis, preservation, and historical research. Moving forward, we recommend further research and development in CV algorithms to expand the scope of architectural style recognition and explore additional applications within the architectural and cultural heritage domains. Additionally, efforts should be made to increase the dataset size, incorporate more architectural styles, and explore the integration of other CV techniques to enhance the model's performance and extend its practical utility.

Acknowledgement and Information Note

The author wishes to thank Fatih Hattatoğlu, PhD., Müslüm Yıldız, MSc., Mustafa Erdoğan, MSc. for the suggestions that helped shape the work and thank the anonymous reviewers for their helpful comments. The paper complies with national and international research and publication ethics. Ethics committee approval was not required for this manuscript.

Author Contribution and Conflict of Interest Declaration Information

The article has a single author and there is no conflict of interest.

References

Boesch, G. (2023a, January 20). *What is Computer Vision? The Complete Tech Guide for 2023 - viso.ai*. Viso.Ai. http://viso.ai/computer-vision/what-is-computer-vision/.

Boesch, G. (2023b, February 21). *Object Detection in 2023: The Definitive Guide - viso.ai*. Viso.Ai. http://viso.ai/deep-learning/object-detection/

Contributors to Wikimedia projects. (2001a, August 8). Casa Batlló - Wikipedia. Retrieved September 25, 2023, from Wikipedia, the free encyclopedia website: https://en.wikipedia.org/wiki/Casa_Batllo

Contributors to Wikimedia projects. (2001b, October 26). Architecture - Wikipedia. Retrieved March 2, 2023, from Wikipedia, the free encyclopedia website: http://en.wikipedia.org/wiki/Architecture

Contributors to Wikimedia projects. (2002a, May 24). Palace of Versailles - Wikipedia. Retrieved September 25, 2023, from Wikipedia, the free encyclopedia website: https://en.wikipedia.org/wiki/Palace_of_Versailles

Contributors to Wikimedia projects. (2002b, July 15). Notre-Dame de Paris - Wikipedia. Retrieved September 25, 2023, from Wikipedia, the free encyclopedia website: https://en.wikipedia.org/wiki/Notre-Dame_de_Paris

Contributors to Wikimedia projects. (2003a, May 6). Milan Cathedral - Wikipedia. Retrieved September 25, 2023, from Wikipedia, the free encyclopedia website: https://en.wikipedia.org/wiki/Milan_Cathedral

Contributors to Wikimedia projects. (2003b, October 6). Trevi Fountain - Wikipedia. Retrieved September 25, 2023, from Wikipedia, the free encyclopedia website: https://en.wikipedia.org/wiki/Trevi_Fountain

Contributors to Wikimedia projects. (2006, May 17). Palladian villas of the Veneto - Wikipedia. Retrieved September 25, 2023, from Wikipedia, the free encyclopedia website: https://en.wikipedia.org/wiki/Palladian_villas_of_the_Veneto

Contributors to Wikimedia projects. (2008, February 18). Object detection - Wikipedia. Retrieved March 10, 2023, from Wikipedia, the free encyclopedia website: http://en.wikipedia.org/wiki/Object_detection

Dwyer, B. (2020, May 8). *When Should I Auto-Orient My Images?* Roboflow Blog; Roboflow Blog. http://blog.roboflow.com/exif-auto-orientation/

Efe, M. O. & Kaynak, O. (1999). A comparative study of neural network structures in identification of nonlinear systems. *Mechatronics*, *3*, 287–300. https://doi.org/10.1016/s0957-4158(98)00047-6

Elmas, Ç. (2018). *Yapay Zeka Uygulamaları* (4th ed.). Seçkin.

Handuo. (2018, August 20). *You only look once (YOLO) -- (1) | Zhang Handuo’s Site*. Zhang Handuo’s Site; Zhang Handuo’s Site. http://zhanghanduo.github.io/post/yolo1/.

Hosni, Y. (2022, October 14). *Overview of Computer Vision Tasks & Applications*. Pub.Towardsai.Net; Towards AI. https://pub.towardsai.net/overview-of-the-computer-vision-tasks-applications-647f63e66e9f

Jocher, G. & Waxmann, S. (2023, May 1). *YOLOv8 - Ultralytics YOLOv8 Docs*. Ultralytics. https://docs.ultralytics.com/models/yolov8/

Jocher, G., Waxmann, S. & Chaurasia, A. (2023, March 12). *Ultralytics YOLOv8 Modes*. Ultralytics YOLOv8 Docs. http://docs.ultralytics.com/#yolo-a-brief-history.

Kasper-Eulaers, M., Hahn, N., Berger, S., Sebulonsen, T., Myrland, Ø. & Kummervold, P. E. (2021). Short communication: detecting heavy goods vehicles in rest areas in winter conditions using YOLOv5. *Algorithms*, *4*, 114. https://doi.org/10.3390/a14040114

Kristo, M., Ivasic-Kos, M. & Pobar, M. (2020). Thermal Object Detection in Difficult Weather Conditions Using YOLO. *IEEE Access*, 125459–125476. https://doi.org/10.1109/access.2020.3007481

Özel, M. A., Baysal, S. S. & Şahin, M. (2021). Derin öğrenme algoritması (YOLO) ile dinamik test süresince süspansiyon parçalarında çatlak tespiti. *Avrupa Bilim ve Teknoloji Dergisi*, *Ejosat*, 1–5. https://doi.org/10.31590/ejosat.952798

Öztürkoğlu, M. (2023a, April 25). *Architectural Buildings3 Computer Vision Project*. Roboflow. http://app.roboflow.com/meryem-dgz60/architecturalbuildings3/1

Öztürkoğlu, M. (2023b, April 30). *Estimating Various Architectural Styles with Computer Vision Methods*. Google Colab. https://colab.research.google.com/drive/1ldJ4P2tMJhCaK7j7LxO-ct3UygW9ERCq?usp=sharing

Öztürkoğlu, M. (2023c, May 9). *Before Train_Estimating Various Architectural Styles with Computer Vision Methods*. Youtube.Com; YouTube. https://www.youtube.com/watch?v=bgctNx_1luE

Öztürkoğlu, M. (2023d, May 10). *Estimating Various Architectural Styles with Computer Vision Methods*. Youtube.Com; YouTube. https://www.youtube.com/watch?v=CC7fakCsCSM

Rath, S. (2023, January 10). *YOLOv8 Ultralytics: State-of-the-Art YOLO Models*. LearnOpenCV. http://learnopencv.com/ultralytics-yolov8/.

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* https://doi.org/10.1109/cvpr.2016.91

Roboflow. (2020, January). Roboflow. https://roboflow.com

Sager, C., Janiesch, C. & Zschech, P. (2021). A survey of image labelling for computer vision applications. *Journal of Business Analytics*,*2*,91–110.https://doi.org/10.1080/2573234x.2021.1908861

Simplilearn. (2022, August 30). What is Epoch in Machine Learning? | Simplilearn. Retrieved September 25, 2023, from Simplilearn.com website: https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-epoch-in-machine-learning?tag=epoch

Su, C. (2008, April 5). *Introduction to Computer Vision*. Carleton.Ca; National Research Council Canada. https://people.scs.carleton.ca/~c_shu/Courses/comp4900d/notes/lect1_intro.pdf

Szeliski, R. (2010). *Computer Vision* (1st ed., p. 5). Springer.

Terven, J. & Cordova-Esparza, Diana-Margarita. (2023). A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond.

Trucco, E. & Verri, A. (1998). *Introductory Techniques for 3-D Computer Vision*. Prentice Hall.

Vitruvius. (1999). *Vitruvius: "Ten Books on Architecture"* (I. D. Rowland, Ed.; T. N. Howe, Trans.). Cambridge University Press.

Williams, K. (2021, July 5). *How to Build a Computer Vision Model*. Medium. http://medium.com/mlearning-ai/what-does-end-to-end-really-mean-f634b193ba00.

Wwymak. (n.d.). *Architecture dataset | Kaggle*. Kaggle: Your Machine Learning and Data Science Community. Retrieved June 28, 2023, from http://www.kaggle.com/datasets/wwymak/architecture-dataset

Xu, Z., Tao, D., Zhang, Y., Wu, J., & Tsoi, A. C. (2014). Architectural Style Classification Using Multinomial Latent Logistic Regression. In *Computer Vision – ECCV 2014* (pp. 600–615). Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-10590-1_39

Yıldız, M. A., Ertosun Yıldız, M. & Beyhan, F. (2023). Developing dynamic and flexible façade design with fractal geometry. *Journal of Architectural Sciences and Applications*, 8 (1), 1-14. DOI: 10.30785/mbud.1230875.