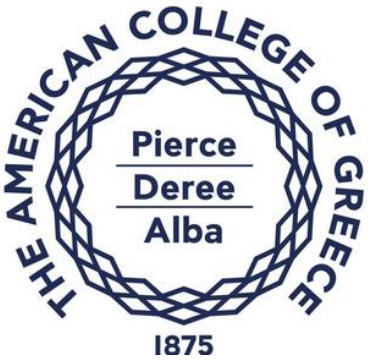
ITC6109A1 - Machine Vision in Data Science

Electric Guitar Body Detection: A Machine Vision Approach



Instructor: Dr. Georgios Drakakis

Submitted by:
Kavvousanaki Evangelia
Kapsalis Christoforos
Proiskos Georgios
Stavrogiannis Christos

Table of Contents

Abbreviations 6

Abstract 7

1. Introduction – Motivation 1

2. Collecting Raw Data – Web Crawler 2

3. Template Matching 3

3.1 Naive Approach 3

3.1.1 Problem Statement 3

3.1.2 End Goal 3

3.1.3 Challenges and Assumptions 3

3.1.4 Steps Implemented 5

3.1.5 Enhanced Template Matching with Image Pyramids and Rotation Augmentation and Keypoint Matching 8

3.2. Template Matching based on Gradient Orientation Features 10

3.2.1 Motivation 10

3.2.2 Methodology for Template Creation 10

3.2.3 Template Matching 11

4. Haar Classifier 13

4.1 Problem Statement 13

4.2 End Goal 13

4.3 Challenges 13

4.4 Steps Implemented 14

5. Results 16

5.1 Naïve Approach 16

5.1.1 Evaluation Metrics 16

5.1.2 Example of Implementation on Stock Image 16

5.2 Enhanced Template Matching 23

5.2.1 Evaluation Metrics 23

5.2.2 Example of Implementation on Stock Image 24

5.3 Haar Classifier – Sample Results 27

6. Conclusions 28
6.1 Template Matching-Naïve Approach 28
6.2 Template Matching-Enhanced Approach 28
6.2 Haar Classifier Approach 28
References 29

Table of Figures

Figure 1: Light Background Case - The most frequent value of the histogram is 255, with a frequency of 70,853 5
Figure 2: Dark Background Case - The most frequent value the histogram is 33, with a frequency of 68,174 6
Figure 3: Filled-in/"Closed" Largest Contour 7
Figure 4: Dilated Canny Edge Result, Ready for Largest Contour Detection 8
Figure 5:Trackbar tool to detect upper body and head in a stratocaster guitar. 11
Figure 7: Similarity Score Considered by Hinterstoisser et al. (2012) 12
Figure 8: Example of Manual Guitar Object Annotation using OpenCV's Annotator 12
Figure 9: Results for bouble_cut template accompanied with confidence on non-rotated test images. 17
Figure 10: Results for single_cut template accompanied with confidence on non-rotated test images. 17
Figure 11: Results for stratocaster template accompanied with confidence on non-rotated test images. 18
Figure 12: Results for telecaster template accompanied with confidence on non-rotated test images. 18
Figure 13: Printed results showing the confidence levels for different transformations (rotation and scaling), comparing the performance of using these transformations versus not using them. The results include the test image and demonstrate the impact of rotation. 19
Figure 14: Results concerning a 3D rotated test image, the highest matched template is the double_cut but second in confidence with small difference was the correct template (single cut). 20
Figure 15: Test image using a combination of keypoints and naive template matching. The results appear below in table 1. 21
Figure 16: Test image using a combination of keypoints and naive template matching. The results appear below in table 2. 21
Figure 17: The ORM performs well, with a high number of well-aligned and evenly distributed keypoint matches. This indicates that the ORM should had successfully recognized and matched the guitar in the test image to the template. However, the template suggested is wrong and as such we consider this effort as unsuccessful. 22
Figure 18: the ORM fails in this case, with some keypoints matching to completely inappropriate points on the test image. This suggests that the ORM is struggling with either the feature extraction or the matching process. Most possibly and as it can be seen above ORB struggle on the curve of the body (upper, lower) since it has no distinct corners. 23
Figure 19: The output showing the best match for a specific template pair. The overlay indicates the matched regions with the highest cosine similarity score. The matched regions are clearly

highlighted, showing the effectiveness of the algorithm in identifying the correct guitar body type. Both the Telecaster Head and Upper-Body Templates were Matched with the Largest Score 26
Figure 20: Example of Match Checking Procedure in Dominant Gradient Orientation Template Matching 26
Figure 21: Small Haar Classifier in Action – its Ability to Only Capture Guitar Necks was Noticeable 27
Figure 22: Message Output after Training the 15th Stage of the Haar Classifier 27

Abbreviations

ST.....Stratocaster

T.....Telecaster

LP.....Les Paul

SG.....Solid Guitar

RGB.....Red-Green-Blue

Abstract

This study focuses on the development of a tool to automatically detect guitar body types in images, motivated by the guitar's cultural significance in visual media. While the association between iconic figures and guitars, particularly the electric guitar, has been well-documented, there is a gap in research concerning the role of guitar body types in shaping consumer perceptions in advertisements. To address this, three distinct approaches are presented regarding the implementation process aimed at detecting guitar body types in photos.

The first approach utilizes a naïve template-matching technique, applying whole-body binary masks to match guitar shapes. The second approach implements an algorithm based on dominant orientations and gradient analysis of local image parts, offering broader applicability across various images. Finally, the prospect of training a Haar classifier to distinguish between different guitar body types is explored, promising enhanced detection accuracy. This tool is intended to facilitate future research on the influence of guitar imagery in advertising and help identify overlooked aspects of music-based marketing strategies.

keywords: guitar body detection, template matching, gradient analysis, Haar classifier, image processing

1. Introduction – Motivation

Celebrity "idols" are often portrayed in iconic stances, like basketball players on the court or actors in famous scenes. For musicians, these images typically feature them performing on stage. Our collective experience and research show that scenes of guitarists holding their guitars leave a lasting impression, especially on the young, making the guitar the preferred instrument for generations (Pelayo, Mallari, & Pelayo, 2015).

Apart from what people find talentful or physically attractive, the electric guitar, as a symbol, has surpassed the levels of the superficial and the boundaries of particular music genres. For decades, the electric guitar has been elevated to a cultural symbol (Han, 2023). This has made it a great audio-visual asset for marketers to use to attract various audiences.

There is extensive research on the associations between celebrities and guitar brands, particularly Fender and Gibson. Scholars like Rayna and Striukova (2018) have explored the different paths these brands have taken in terms of guitar characteristics and marketing. However, the physical characteristics of guitars as objects and their impact on consumers of guitar-themed content seem to have been overlooked.

Thus, in the current study, we intend to contribute to studying the impact of the incorporation of guitar body types in advertisements on consumer groups. We will do this by attempting to develop a tool that can be used to detect guitar body types in photos automatically. Such a tool and the facilitation of subsequent research could help uncover weaknesses in how the advertising and media industries approach various consumer groups with music-based content.

Our attempt followed three phases, utilizing methods and algorithms of increasing sophistication, difficulty of implementation, and generalization capabilities. The initial approach was based on a naïve technique utilizing whole-body binary mask template matching. Then, we tried to implement an algorithm based on the dominant orientations and gradients on local image parts, promising broad applicability. Last, we considered the prospect of developing, training, and implementing a Haar classifier to detect different guitar bodies in images.

2. Collecting Raw Data – Web Crawler

The first milestone towards our objective was deciding on a set of images to use in testing pre-processing methods. Some of our team members' tenure in musicianship quickly pointed at online musical instrument marketplaces to find pictures of guitars that are of high quality and "clean" enough for our -at least initial- algorithms to work with.

We decided to make our algorithms able to work on specific guitar body types, and specifically the hallmark models offered by the giants of Fender and Gibson, which have also been imitated by a multitude of highly popular brands. These body types are the "Stratocaster-ST", "Telecaster-T", "Les Paul-LP", and "Solid Guitar-SG" guitars. A short survey of various online music stores led us to thomann.de (Thomann GmbH, n.d.), due to the observed consistency of the conditions under which guitar sample photos were taken in terms of lighting, guitar orientation, and shadows, among other factors.

For the quick retrieval of a wide array of guitar photos of the different types, we developed a web crawler using the Python implementation of the Playwright library (Microsoft, n.d.). This was facilitated by the consistency of the HTML structure of Thomann's pages, which allowed us to rapidly develop a script to collect images, aiming at creating a set of one hundred images per type.

The relevant image files (of the ".jpg" format) were handled locally on our machines.

3. Template Matching

3.1 Naive Approach

3.1.1 Problem Statement

This solution essentially consists of an object detection system of some type, which is concentrated on guitar objects but can also distinguish between different types of these objects of interest.

We quickly realized that even stock photos from Thomann varied a lot in their colors, the background, as well as their structure and shape, quickly **"taking out of the picture"** any solution incorporating simple thresholding approaches or considering the local gradients of pictures showcasing guitars to create generalized templates.

However, as a first step of more of an exploratory nature, we deemed it would be helpful to develop a rather simple system designed to detect guitar body types under ideal conditions, like the guitar stock photos we scraped from Thomann.

3.1.2 End Goal

The north star of this implementation was the creation of generalized one-piece and full-body templates that would numerically represent and efficiently recognize the main features that make out an "ST" versus a "T", a "SG" or an "LP" guitar in an automated manner. These templates were designed with capturing a "definitive" whole-body shape of the guitars of each type.

In other words, our guiding principles were:

- Broad applicability
- Automation
- Numeric expression of Guitar Body-Distinguishing Feature that go beyond what our experience suggests

This way, we will have developed a prototype of a system building on top of simple methods of machine vision to provide tangible value to various sectors.

3.1.3 Challenges and Assumptions

Fundamental Assumption

This naïve approach is built upon a very restraining assumption that there is indeed a "definitive" shape for the types of guitars considered. This proposition could be contested, as there are often variations even between the same body-type models of a particular brand. Furthermore, we decided to only consider right-handed guitars for this approach, as we deemed that the insertion of a step to detect the inverted of left-handed guitars would distract us from the complexity of our goal at that stage.

Image Size, Scaling, Rotation, and Lighting Conditions

The visual observation of hundreds of guitar images from Thomann pointed to the size, scaling, rotation and lighting conditions on each of them being consistent. Nonetheless, this was not guaranteed and posed a factor out of our control, so we had to assume it to be true.

Background Variations

The photos retrieved from Thomann had consistent backgrounds, yet the colors of the latter changed in an unpredictable manner. Moreover, there were cases of guitars with body colors very similar to that of the background, translating in very similar intensity values. Even in such cases, since the guitar only covered a small proportion of the image's surface and the background color showed no shadows or glitches, the implementation of band thresholding to capture the background and drop it to isolate the guitar bodies seemed feasible.

In this direction, we initially adopted a similar pre-processing approach that was based on the most frequent value of an image's histogram. After observing the results of this analysis for multiple images, we concluded that only two distinct cases existed. However, after experimentation with later pre-processing steps, we decided that this step was not needed and opted for the approach detailed in Section 3.1.4.

Anomalous Images

We noticed cases of guitars where our crawler failed to retrieve the "front" image of the instrument, possibly due to Javascript functions preventing it from doing so. This resulted in our corpus of images containing rotated images, or images of left-handed guitars. We manually tracked these problematic images and replaced them to ensure consistent conditions for our template creation process.

3.1.4 Steps Implemented

Considering only the Luminance Channel of Each Image

We imported every image in grayscale to reduce the complexity of the work by processing only a single-color channel. This was very conveniently implemented utilizing methods of the OpenCV (Bradski and Kaehler, 2000) library for the Python programming language (Python Software Foundation, n.d.).

Histogram-Based Pre-Processing

As mentioned in Section 3.1.3, we detected two cases of images based on the intensity values of the luminance channel of their backgrounds. Pre-processing on them would be different to ultimately provide their contributing mask to the corresponding generalized template of their guitar type. As a result, we decided to monitor the cases under which they fall based on their most frequent intensity value (as shown in Figure 1 and 2) so we can conveniently implement different operations on each case.

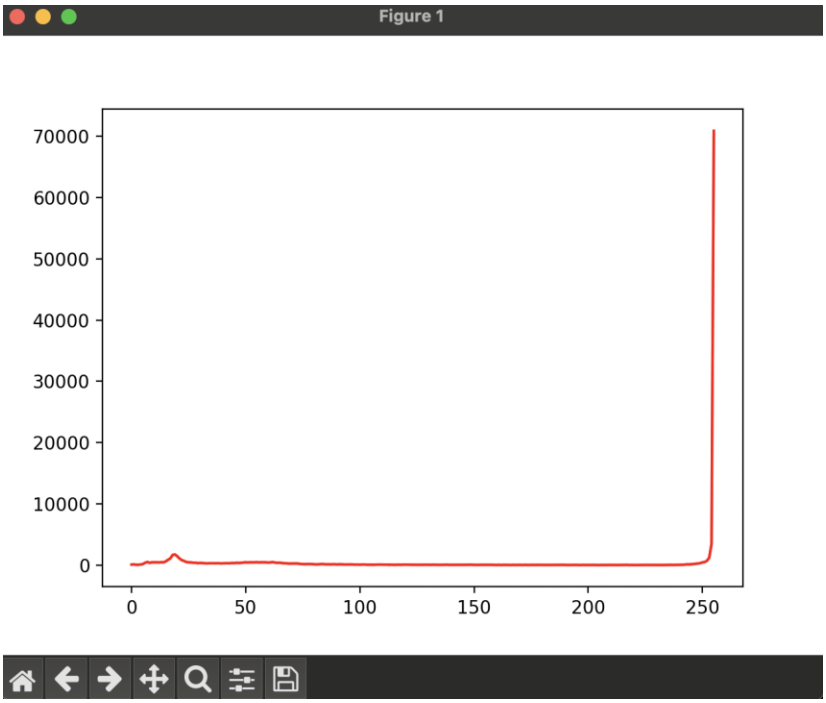


Figure 1: Light Background Case - The most frequent value of the histogram is 255, with a frequency of 70,853

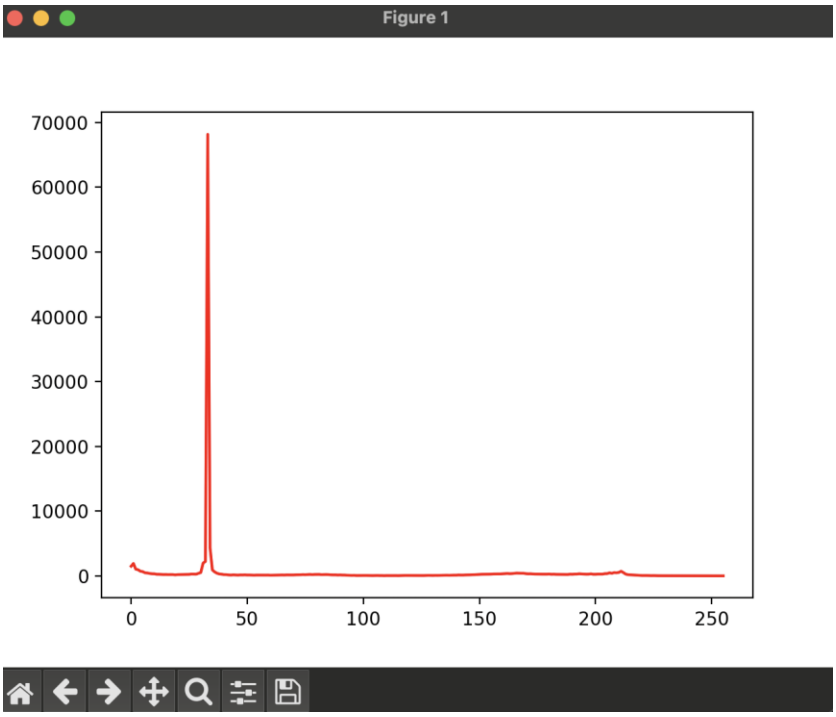


Figure 2: Dark Background Case - The most frequent value the histogram is 33, with a frequency of 68,174

Moreover, the experimentation with multiple images showcased the possibility of some images of the 'dark background' case having only a middle 'dark' area, surrounded by white values, falsely skewing the results of the histogram operations. Thus, we found two central 'columns' containing the actual 'guitar' part, as shown in Figure 4.

Image Blurring

To prevent the (empirically unlikely) scenario of Thomann images containing noise, we implemented Gaussian blurring with a small standard deviation value for the underlying theoretical normal distribution (sigma value equal to 1) to minimize its effect on edge smoothing while contributing to noise reduction.

Edge Detection – Keeping only the Largest Contour

Since the image backgrounds proved to be exceptionally consistent, we implemented Canny edge detection. We preferred this against the more computationally efficient Sobel detector to capture

edges in detail. This was based on our trust in that, after some rounds of dilation, the largest contour would match the guitar's outline, like shown in Figure 4.

We contemplated implementing component analysis in this stage, but since we were interested solely in the largest contour suitable to our guitar shape analysis (OpenCV, n.d.), we decided that discarding noise particles would not provide for any value. To ensure the resource-lightness of this step, we used the "CHAIN_APPROX_SIMPLE" function for contour approximation.

An interesting remark here is that the main threshold value for the images with dark background was optimized to be lower than that of light background ones. This was because dark-colored guitars were more likely to approximate the total black (intensity value of 0) than light-colored guitars to approximate the absolute intensity (value of 255).

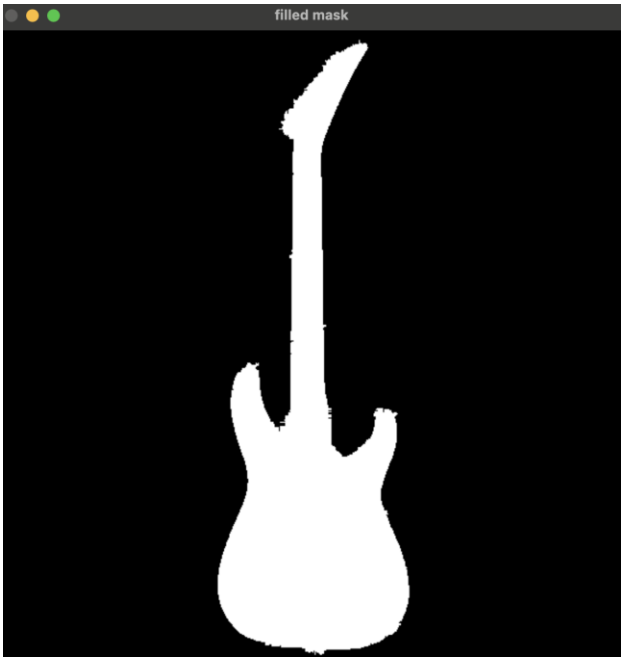


Figure 3: Filled-in/"Closed" Largest Contour

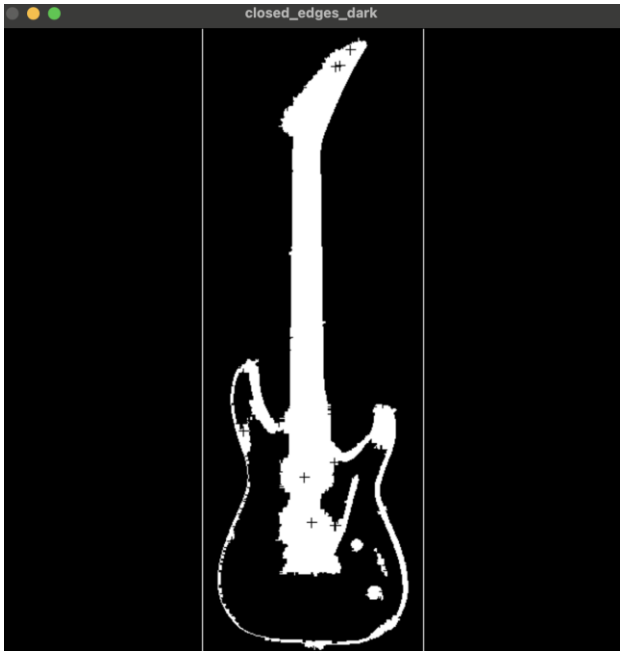


Figure 4: Dilated Canny Edge Result, Ready for Largest Contour Detection

Morphological Operations

After observing that the previous steps successfully detected the outline of guitars, the completion of the mask generation process only required a closing operation to be implemented, consisting of multiple rounds of dilation to fill in gaps in our outline which are then followed by an equal number of erosion steps to return the object of interest back to its initial size. The "drawContours" function of OpenCV (n.d.) conveniently implemented the previous, resulting in a binary image in which information (i.e. the guitar) is highlighted in white, as in Figure 3.

Template Generalization

For each guitar type, we stacked and averaged the masks extracted for each guitar belonging to it to create the "definitive" template for that type, based on our assumptions.

3.1.5 Enhanced Template Matching with Image Pyramids and Rotation Augmentation and Keypoint Matching

This part of the report presents an exploration of template matching with various transformations, including rotation, scaling, and multi-scale matching using an image pyramid approach. Additionally, we examine the role of keypoint matching using the ORB algorithm to improve precision when confidence scores between template matches are similar.

Template Matching with Transformations

Templates were matched to a test image using several transformations designed to handle different variations in orientation and scale. These transformations included both 2D and 3D rotations, as well as scaling through an image pyramid approach.

- **2D Rotation (z-axis)**: The templates were rotated around the z-axis in 5-degree increments, from 0° to 360°. The rotate2D function from OpenCV was used to perform the rotation.
- **3D Rotation (x and y axes)**: In addition to 2D rotations, 3D rotations were simulated by rotating the templates along both the x and y axes at specific angles (0°, 45°, 90°, and 180°). This extended the testing to more complex transformations, simulating scenarios where objects may be rotated in 3D space.

Image Pyramid for Multi-Scale Matching

To address scale variations between the test image and the template, we used an image pyramid approach. The image pyramid involved resizing both the test image and the templates to multiple scales ranging from 0.2 to 2.0. This approach allowed us to test template matching across different scales, ensuring that the algorithm could detect templates regardless of their size relative to the test image.

Best Match Selection

For each transformation applied (rotation and scaling), the template similarity (using cv2 libraries) between the rotated or scaled templates and the test image was calculated. The template with the highest match (confidence) score was selected as the best match.

Keypoint Matching with ORB

In cases where the confidence scores between templates were close and traditional template matching process did not yield distinct results, we employed ORB (Oriented FAST and Rotated BRIEF) keypoint matching to refine the results and improve precision. ORB is a feature detection and description algorithm that is particularly effective in handling images that have undergone transformations developed by CV lab (OpenCV, n.d.; Rublee et al., 2011). By focusing on keypoints and descriptors, ORB allows for a more robust and accurate match, especially when the match scores between templates are close.

3.2. Template Matching based on Gradient Orientation Features

3.2.1 Motivation

The results of the approach discussed in Section 3.1 were poor in random images, due to the high variance in not only outer conditions (e.g. the lighting conditions or the scaling and rotation of a guitar) but also the characteristics of particular guitar models, like their colors and their relationship to their surroundings, the type of their body edges (sharp versus smoothed), custom guitar head models, and endless possibilities of customizations. This made us look for template matching-related applications that are color- and contrast- agnostic. Our research resulted in a very limited number of relevant attempts based on template matching and not machine or deep learning techniques.

The main sources of inspiration were the papers by Hinterstoisser et al. (2012) and Daftry et al. (2021). It was primarily the latter that drove us to experiment with enhanced template matching approaches before considering pure machine learning-based techniques. And that was because it promised exceptional results in a critical task like the localization of sample tubes on the Martian surface using a –seemingly- outdated approach.

3.2.2 Methodology for Template Creation

Both these papers suggested that, for texture-less objects (such as the typical case of a guitar), the gradient orientations from the objects' silhouette/outline suffices as a feature to extract for template matching in a computationally efficient and color- and contrast-agnostic manner.

For each type of guitar, our previous experience pointed at two features serving as the most effective hints at per the body type: the head of the guitar, and the upper half of its body. Thus, we used the previous approach to build generalized gradient orientation-based templates for the head and upper body of each guitar, by averaging the features extracted from every related guitar. These features were weighted by the corresponding gradient values so as to ensure that we are capturing actually the outline sides of these parts.

The previous was deemed promising due to our corpus of images and previous analysis facilitating the easy extraction of guitar outlines and the examination of the gradient orientations on them.

Following the implementation of Hinterstoisser et al. (2012), we parsed each of the images used for template creation without converting it to grayscale. By retaining their initial color channels (three in number, as the images were represented based on the Red-Green-Blue/RGB color model),

we parsed them pixel by pixel and computed the gradients on each color channel. The feature representing every given pixel was the gradient orientation on the color channel with the largest gradient. Additionally, we experimented with various pre-set coordinates for detecting the guitar head and upper body in our optimized "training" images, as shown in Figure 5. For both training and input images, the only pre-processing step implemented was Gaussian blurring (on each color channel separately) to prevent noise from skewing our results.

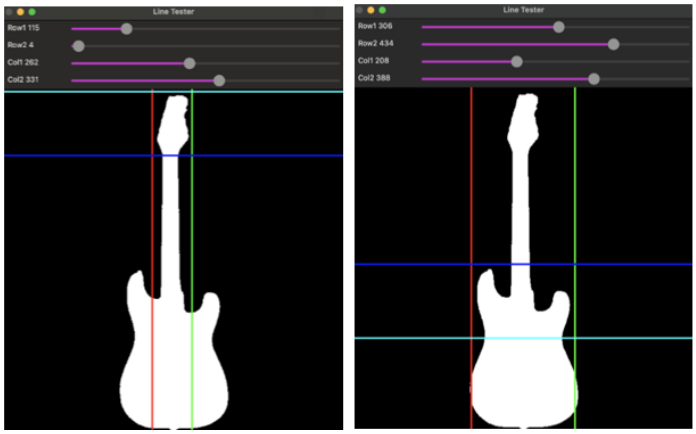


Figure 5:Trackbar tool to detect upper body and head in a stratocaster guitar.

3.2.3 Template Matching

For template matching, we retained the same considerations pertaining to image pyramids and rotation augmentation as discussed in Section 3.1.5, but in this case both templates of a particular guitar body type had to match on an input image to consider it of that type. To calculate a total "score" of a particular image, we calculated the average of its matching confidence with the head and the upper-body templates.

The similarity score utilized matched that of the Hinterstoisser et al. (2012) implementation and is showcased in Figure 7.

2.1 Similarity Measure

Our unoptimized similarity measure can be seen as the measure defined by Steger in [18] modified to be robust to small translations and deformations. Steger suggests to use:

$$\mathcal{E}_{\text{Steger}}(\mathcal{I}, \mathcal{T}, c) = \sum_{r \in \mathcal{P}} |\cos(\textbf{ori}(\mathcal{O}, r) - \textbf{ori}(\mathcal{I}, c + r))| \ , \tag{1}$$

where $\textbf{ori}(\mathcal{O}, r)$ is the gradient orientation in radians at location r in a reference image $\mathcal{O}$ of an object to detect. Similarly, $\textbf{ori}(\mathcal{I}, c+r)$ is the gradient orientation at c shifted by r in the input image $\mathcal{I}$. We use a list, denoted by $\mathcal{P}$, to define the locations r to be considered in $\mathcal{O}$. This way we can deal with arbitrarily shaped objects efficiently. A template $\mathcal{T}$ is therefore defined as a pair $\mathcal{T} = (\mathcal{O}, \mathcal{P})$.

Figure 6: Similarity Score Considered by Hinterstoisser et al. (2012)

Figure 7: Example of Manual Guitar Object Annotation using OpenCV's Annotator

4. Haar Classifier

4.1 Problem Statement

The initial frustration caused by the inefficacy of our simplistic template matching implementation quickly turned our interest to machine learning-based approaches. The Haar classifier, originally developed for face-detection applications, posed an attractive solution to cover for the weakness of our previous setup.

Just as human faces bear universal characteristics which can more or less differ from face to face, the same is true for electric guitars. A guitar of such needs a head with its string tuners, frets, pickups, a bridge, and volume knobs to operate. These features show significant variance among different guitars. Still, their general outline and relative positions seem predictable, making them promising candidate objects for this method. This argumentation justified the suitability of this approach among our team members.

4.2 End Goal

In this approach, then, we sought to determine whether a Haar classifier, originally devised for tasks like face detection, could be adapted to identify guitar body types. The problem lies in adapting the algorithm to capture the diversity of guitar shapes while ensuring computational efficiency and accuracy, given our limited training dataset.

To this end, our first milestone was to train a Haar classifier that could detect guitars in images. This was essential before extending the classification system to also account for the body type of a detected guitar. The latter was not implemented in this project, due to the insurmountable difficulties that came up in the implementation of the former.

4.3 Challenges

No Training nor Testing Datasets Available

Unlike large-scale datasets available for face detection, we were unable to find any training dataset containing annotated guitar objects.

Visual Complexity of Guitars

The variations of guitar features are more nuanced than that of human face characteristics. This could make the successful training of such a system require large-scale training and testing datasets that are highly variable, and meticulously optimized.

Limited Computational Resources: We were only able to execute a local implementation of this approach, allowing only for a short training period of our algorithm.

4.4 Steps Implemented

Dataset Preparation

1. Our training data ("positive images" from now on) comprised of manually annotated images using the OpenCV Haar Annotation Tool (OpenCV, n.d.), as shown in Figure 13.
2. Our testing data ("negative images") consisted of diverse negative samples drawn from stock image websites, like Figure 14. We opted for images featuring objects that could be confuse a naïve algorithm to be guitars, like trees, hockey sticks, and tennis rackets.
3. All images were converted to the same size (600x600, to match all images retrieved from Thomann), and we only considered their luminance channel.
4. All in all, we considered 300 positive and 50 negative images, and a sliding window of 24x24 pixels. We only allowed for 15 stages of training for our classifier, due to our limited resources and the high computational load it caused to our machines compared to other conventional methods.

Training with AdaBoost – Default Cascade Classifiers Structure

5. We sticked to the implementation of Viola and Jones (2001) which only utilized the luminance channel to allow for rapid detection of objects by training the cascade of classifiers on the AdaBoost algorithm to eliminate non-guitar regions.
6. This matches the implementation of OpenCV based on a highly-esteemed GitHub code-base (Mrnugget, n.d.).

Performance Metrics

Prioritizing true positives, we based our classifier evaluation on the false negative rate.

Preprocessing for Detection on Input Images

7. Images were converted to grayscale to standardize input data and align with the algorithm's original design.
8. Gaussian smoothing was applied to reduce noise without compromising critical features.
9. Edge detection and morphological operations helped refine detected regions for subsequent processing.

5. Results

5.1 Naïve Approach

5.1.1 Evaluation Metrics

In the template matching naïve approach involving transformations both regarding rotation and scaling, several evaluation metrics were used to assess the effectiveness of this matching technique as discussed below.

Confidence Level: The primary metric for assessing the similarity between the template and the test image. The confidence level indicates how closely the template matches the test image, with higher values representing a better match. This metric provides a quick way to gauge the overall performance of template matching.

Keypoint Matching Score: In cases where the confidence levels between rotated or scaled templates were close, keypoint matching using ORB (Oriented FAST and Rotated BRIEF) was employed to improve precision. By detecting and matching keypoints between the template and test image, this technique ensures a more accurate and reliable comparison, especially when transformations cause subtle variations in the images.

Regarding the keypoint matching approach used the number of matches to distinguish close confidence scores. A higher **number of matches** indicated better performance, suggesting successful feature identification. **Match accuracy**, especially well-aligned features (e.g., headstock), confirmed correct matches. Lastly, **evenly distributed matches** across the entire guitar image indicated a robust matching process.

In conclusion, the above metrics provided a comprehensive evaluation of the template matching process, helping identify the most effective transformations and ensuring reliable results in different matching scenarios.

5.1.2 Example of Implementation on Stock Image

Execution in Ideal Scenarios

In our initial tests without rotation and under optimal conditions, the script performed exceptionally well. Keypoint matching consistently produced accurate results, with templates aligning perfectly to test images. Confidence scores were high, and matches were clear,

demonstrating the script's effectiveness in ideal scenarios without rotational or environmental variations.

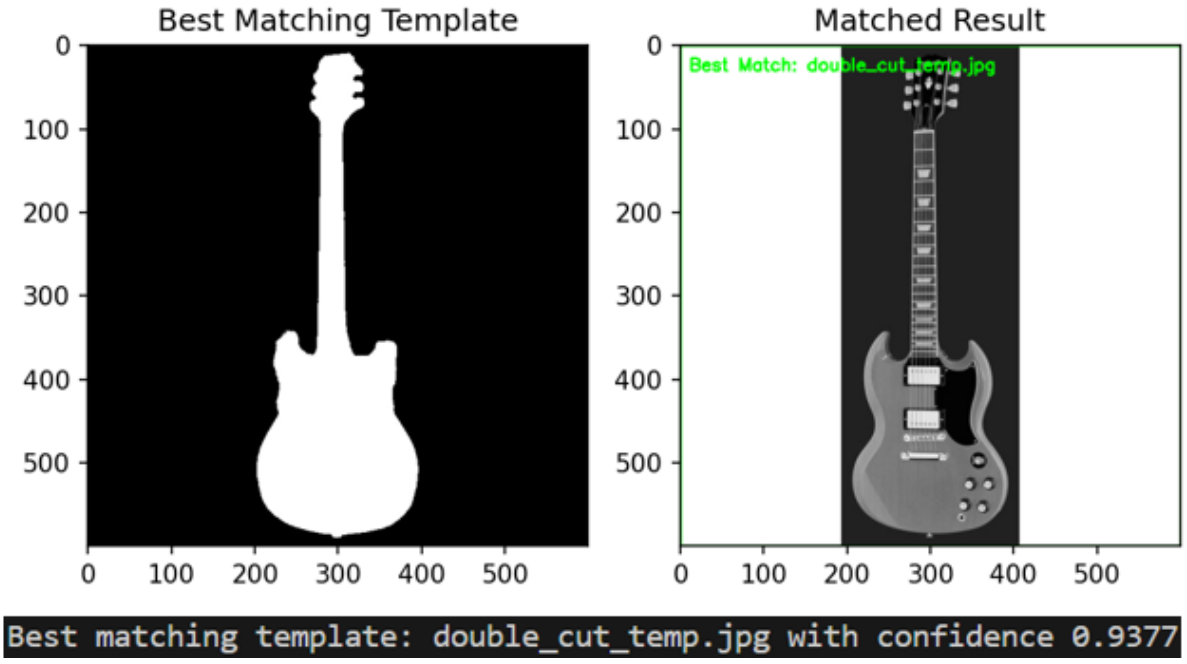


Figure 8: Results for bouble_cut template accompanied with confidence on non-rotated test images.

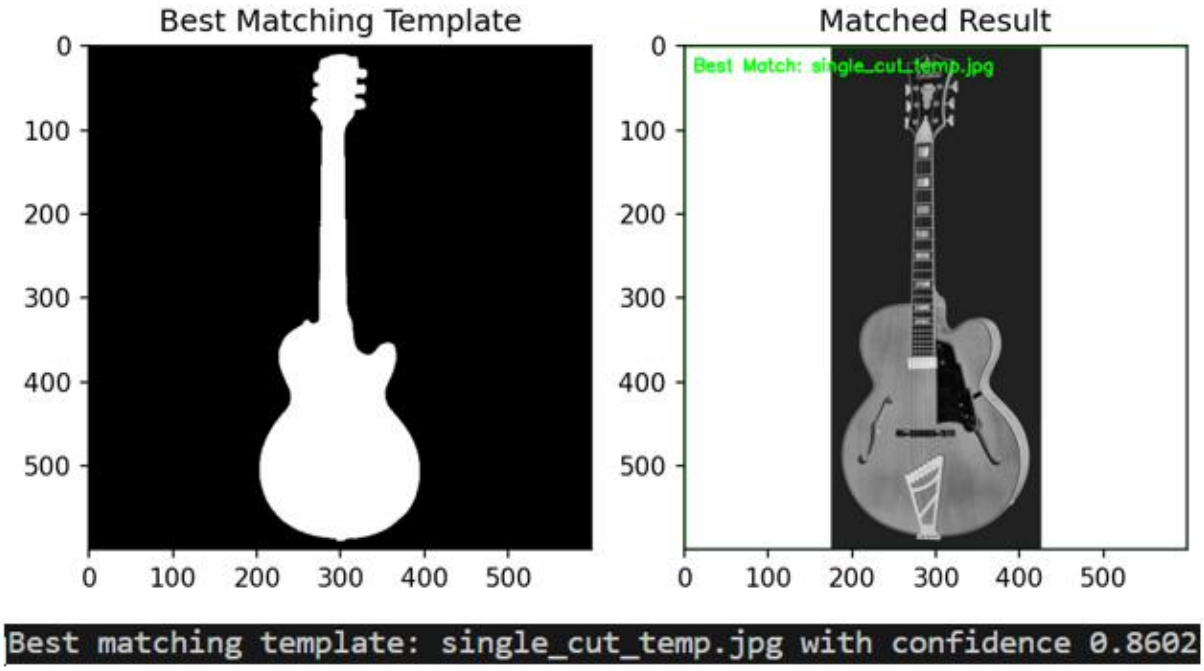


Figure 9: Results for single_cut template accompanied with confidence on non-rotated test images.

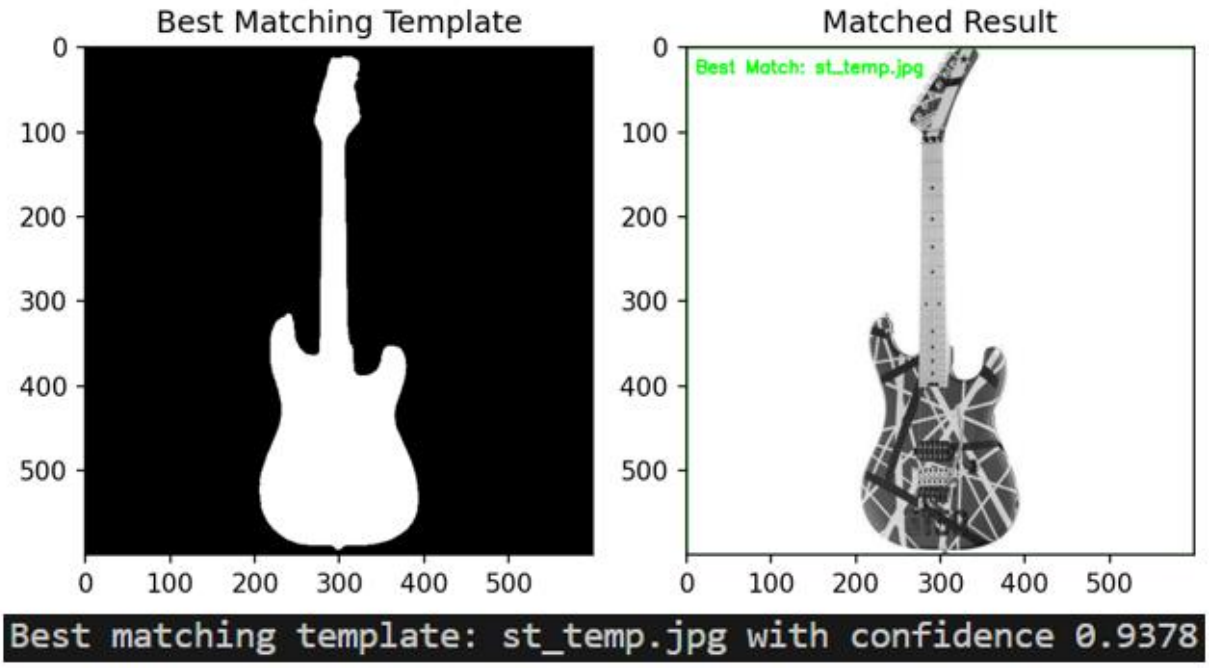


Figure 10: Results for stratocaster template accompanied with confidence on non-rotated test images.

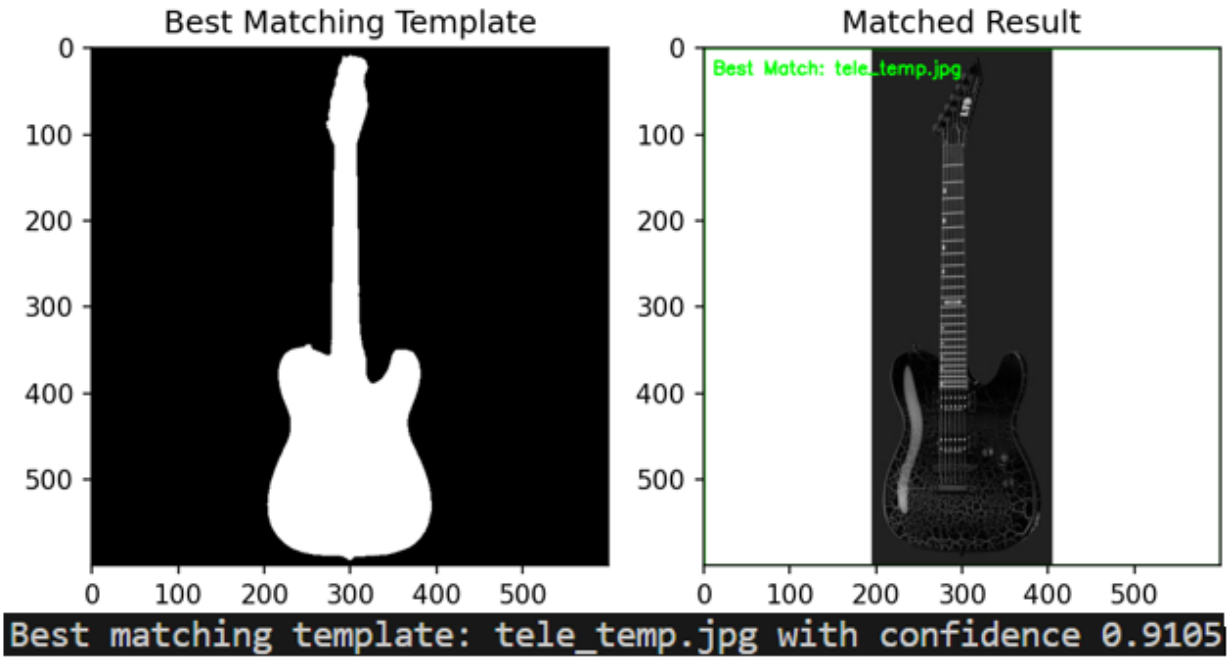


Figure 11: Results for telecaster template accompanied with confidence on non-rotated test images.

Rotation and Confidence Comparison

The rotation trials rotated templates by 5° increments (0° to 360°). Best matches occurred when template and test image orientations closely aligned, though several angles had similar scores. Simulated 3D rotations (0°, 45°, 90°, 180°) added variety but did not significantly improve match accuracy over 2D rotations. The image pyramid method effectively handled scale variations, with the best matches found when template size matched the object in the test image. Extreme scales also performed well, with a range of 0.2 to 2.0 in steps of 0.1. Visualization of results include overlays of matched templates and details like confidence scores, angles, and scales as shown in Figure 13.

Although ORB should have improved the accuracy in some instances (by counting highest number of matches in each case), the results did not dramatically alter the overall performance and visualizing the results.

```
Processing test image: ./irregulars/aaaa.jpg
Best match found at location (0, 0) with score 0.973717987537384
Template: ./templates/st_temp.jpg, Rotation: 90, Scale: 1.0
```

```
Processing aaaa.jpg...
['./templates/double_cut_temp.jpg', './templates/single_cut_t
es/tele_temp.jpg']
Best matching template: st_temp.jpg with confidence -0.1143
```

Figure 12: Printed results showing the confidence levels for different transformations (rotation and scaling), comparing the performance of using these transformations versus not using them. The results include the test image and demonstrate the impact of rotation.

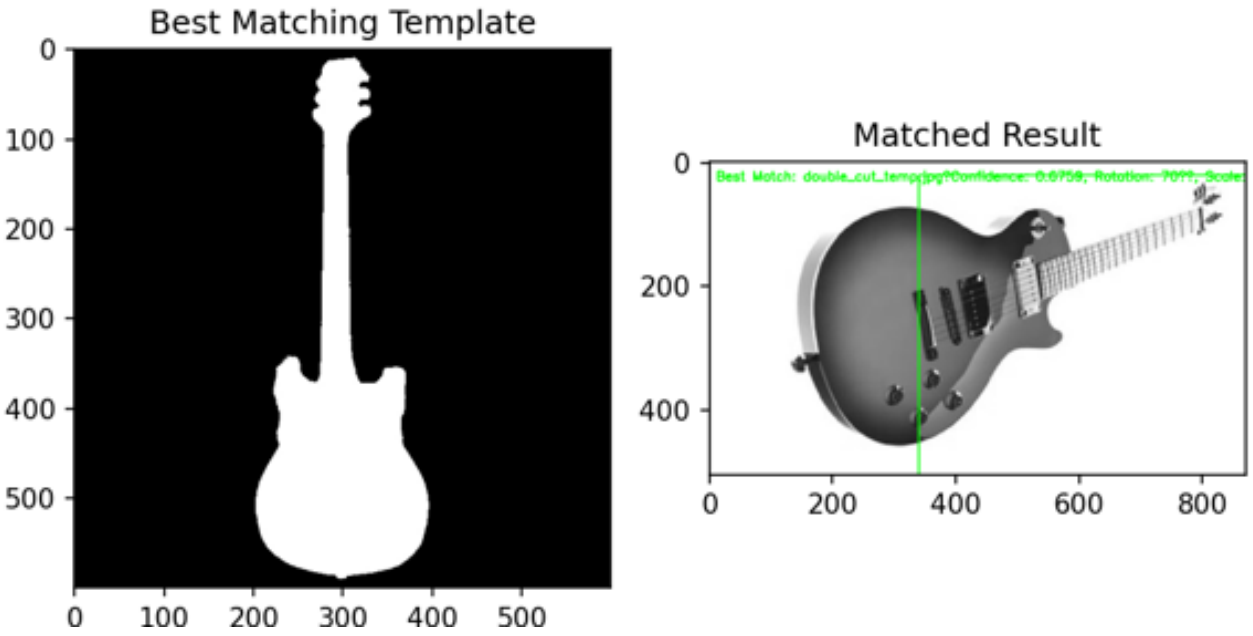


Figure 13: Results concerning a 3D rotated test image, the highest matched template is the double_cut but second in confidence with small difference was the correct template (single cut).

Keypoint Matching Analysis

We used Object Recognition Model (ORM) in matching keypoints between a test image of a guitar (single_cut_temp.jpg) and all the possible templates.

The keypoint matches are visualized by lines connecting corresponding features between the test image and the template. Upon examining the results we observe that the ORM demonstrates good performance. Moreover, even in cases where the results were correct, the incorrect keypoint matches as can be seen in Figure 18. By further examining the incorrect matches and considering improvements in preprocessing and feature extraction we derive that the overall performance of the keypoint matching process was not optimal for all the cases we applied it on and should have been enhanced. This effort was abandoned based on the inconsistent nature of our results.

Possible Reasons for Incorrect Matches

Incorrect matches may stem from several factors related to rotation and scaling in irregular images:

Feature Ambiguity: Similar features in the test image (e.g., upper and lower guitar body curves) could be confused, leading to mismatches.

Template Quality: If the template lacks distinct features or corners, the algorithm may struggle to find accurate matches.

One improvement that could have been applied is **focusing on the head of the template** and test image since it has distinct features (corners) and is quite unique in order to distinguish between

guitar types especially considering that the keypoint matching approach was implemented as a supplementary technique in case we had a test image matching very closely (based on the confidence level) two different templates.

Nevertheless, when implementing this naive approach, we **hadn't yet though of matching distinct parts of the guitar**, and we were working on matching the whole-body-template.

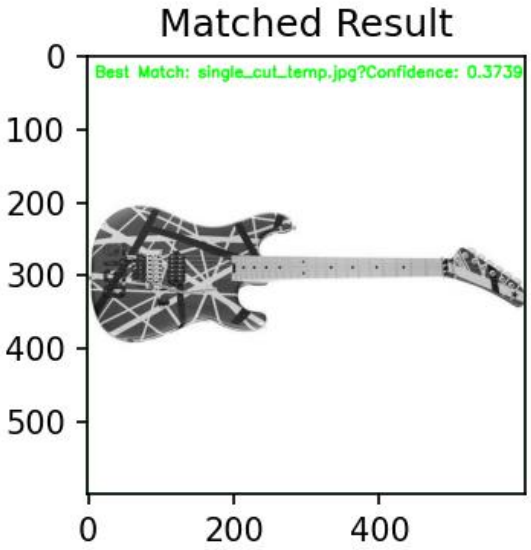


Figure 14: Test image using a combination of keypoints and naive template matching. The results appear below in table 1.

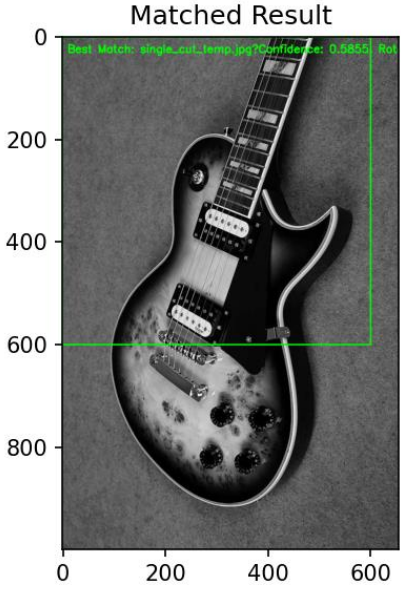


Figure 15: Test image using a combination of keypoints and naive template matching. The results appear below in table 2.

Table 1: Results for figure 7 combining both keypoints and confidence score and showcasing the best match.

Table 2: Results for figure 8 combining both keypoints and confidence score and showcasing the best match

Template	Template Matching	Rotation	Scale	Num of keypoints	Final Score
double_cut_temp.jpg	0.3814	80°	1.00	39	19.6907
single_cut_temp.jpg	0.3739	80°	1.00	42	21.1869
st_temp.jpg	0.3796	85°	1.00	32	16.1898
tele_temp.jpg	0.3882	85°	1.00	29	14.6941
Best Template: single_cut_temp.jpg	0.3739	80°	1.00	42	21.1869

Template	Template Matching	Rotation	Scale	Num of keypoints	Final Score
double_cut_temp.jpg	0.5724	175°	1.00	46	23.2862
single_cut_temp.jpg	0.5855	195°	1.00	59	29.7927
st_temp.jpg	0.5937	175°	1.00	39	19.7968
tele_temp.jpg	0.5997	175°	1.00	28	14.2998
Best Template: single_cut_temp.jpg	0.5855	195°	1.00	59	29.7927

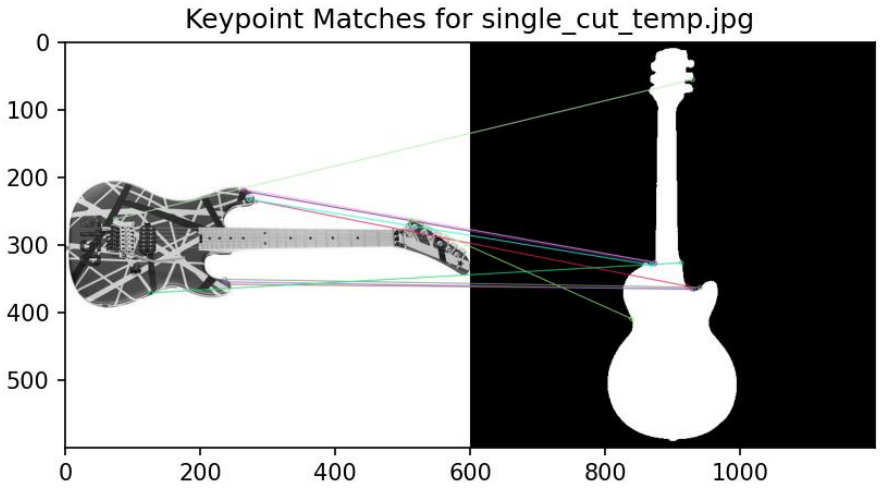


Figure 16: The ORM performs well, with a high number of well-aligned and evenly distributed keypoint matches. This indicates that the ORM should had successfully recognized and matched the guitar in the test image to the template. However, the template suggested is wrong and as such we consider this effort as unsuccessful.

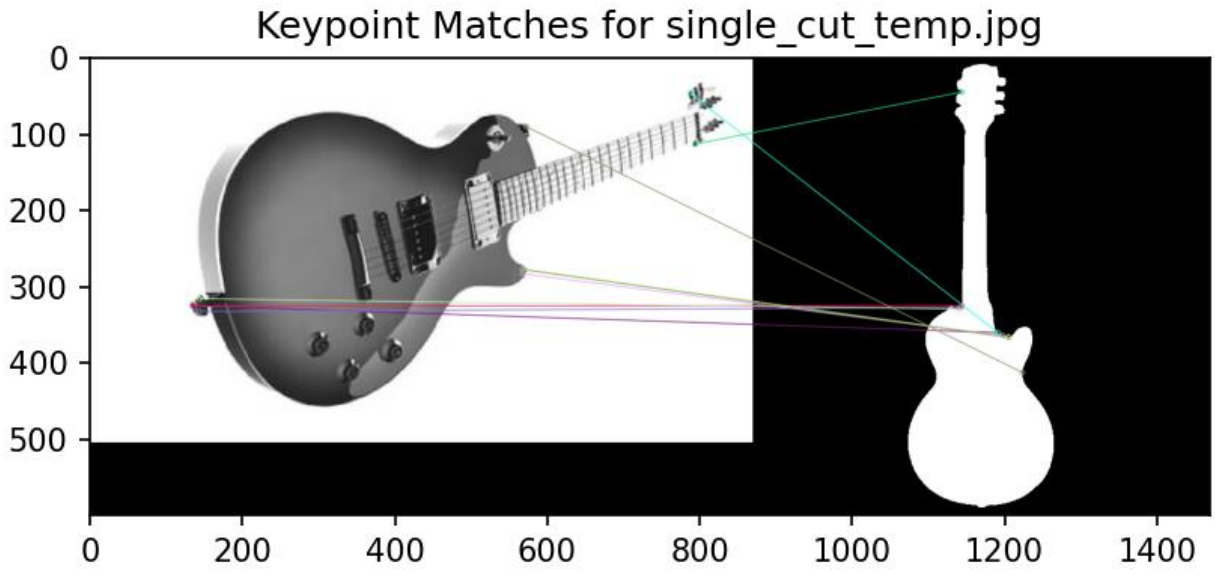


Figure 17: the ORM fails in this case, with some keypoints matching to completely inappropriate points on the test image. This suggests that the ORM is struggling with either the feature extraction or the matching process. Most possibly and as it can be seen above ORB struggle on the curve of the body (upper, lower) since it has no distinct corners.

5.2 Enhanced Template Matching

5.2.1 Evaluation Metrics

Our second approach in the development of a tool for detecting guitar body types in images involves an algorithm based on dominant orientations and gradient analysis of local image parts. This part of the report discusses the evaluation metrics used to assess the performance of our implementation.

Cosine Similarity

Cosine similarity is a metric used to evaluate the match between the template and the target image. This metric measures the cosine of the angle between two non-zero vectors, which in this case are the gradient orientations of the template and the target image. The cosine similarity is calculated as follows:

cos_similarity = np.sum(np.cos(np.radians(diff_)))

where **diff_** is the difference between the gradient orientations of the template and the region of interest (ROI) in the target image. The scored ranges from -1 to 1, with higher cosine similarity scores indicate better matches between the template and the target image.

Quantized Orientations

Gradient orientations are quantized into bins (default: 8) to simplify data and improve efficiency. Bin width is:

bin_width = (2 * np.pi) / bins

The quantized orientations are then used to compare the dominant orientations in the template and the target image.

Thresholding

Matches are determined using a threshold (default: 0.885), creating a binary mask to focus on significant matches.

match_mask = (result >= threshold).astype(np.uint8)

Edge Detection and Preprocessing

Edges are detected after Gaussian blurring to reduce noise, enhancing gradient features for more robust matching.

Normalization

The gradient magnitudes are normalized to the range [0, 255] which helps comparing the gradient orientations more effectively using the following formula

magnitude = cv2.normalize(magnitude, None, 0, 255, cv2.NORM_MINMAX)

Overall, this approach shows higher accuracy, especially in simple scenes, by combining cosine similarity, quantized orientations, thresholding, rotation invariance, edge detection, and normalization.

5.2.2 Example of Implementation on Stock Image

Results

The second approach employed an algorithm based on dominant orientations and gradient analysis of local image parts to detect guitar body types. The process involved several key steps, from image preprocessing to template matching and rotation analysis.

Process Overview

We began by rotating the template images to various angles to achieve rotation invariance. This step was crucial for ensuring that the algorithm could accurately detect guitar body types regardless of their orientation in the target image. The rotation angles used included 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°.

Furthermore, we computed the dominant gradient orientations for both the template and target images. This involved calculating the gradient magnitudes and orientations for each pixel and determining the dominant orientation for each local region. The gradient orientations were then quantized into a specified number of bins to simplify the orientation data and make the matching process more efficient. This quantization helped in reducing the complexity of the orientation data and achieved faster matching.

The target image underwent preprocessing, which included Gaussian blurring to reduce noise and Canny edge detection to highlight the edges. The detected edges were used to compute the dominant gradient orientations, which were then normalized to ensure consistent values across the image. This preprocessing step was critical in enhancing the gradient features and making the algorithm more robust to variations in lighting and background clutter.

The core of the matching process involved comparing the quantized orientations of the template and target images. The script calculated the cosine similarity between the orientations to determine the match quality. Thresholding was applied to the cosine similarity scores to filter out weak matches and focus on significant ones. The match mask was created to highlight the regions with the highest similarity scores, indicating the best matches.

Results and Observations

The algorithm was tested on various template pairs and target images, including those with suboptimal conditions. Despite the challenges posed by variations in lighting, angles, and background clutter, the algorithm demonstrated a remarkable ability to identify the best rotation angle for each template.

The use of dominant gradient orientations and quantized orientations allowed the algorithm to handle different angles and partial occlusions effectively.

Quantitative Results

The implementation performance was evaluated using several metrics, including the cosine similarity score and the number of correctly matched keypoints. The cosine similarity score provided a measure of the match quality, with higher scores indicating better matches. The number of correctly matched keypoints indicated the accuracy of the algorithm in identifying the guitar body type.

In most test cases, the algorithm achieved high cosine similarity scores, indicating accurate matches between the template and the target image. The rotation invariance feature ensured that the algorithm could handle different orientations, further enhancing its robustness.

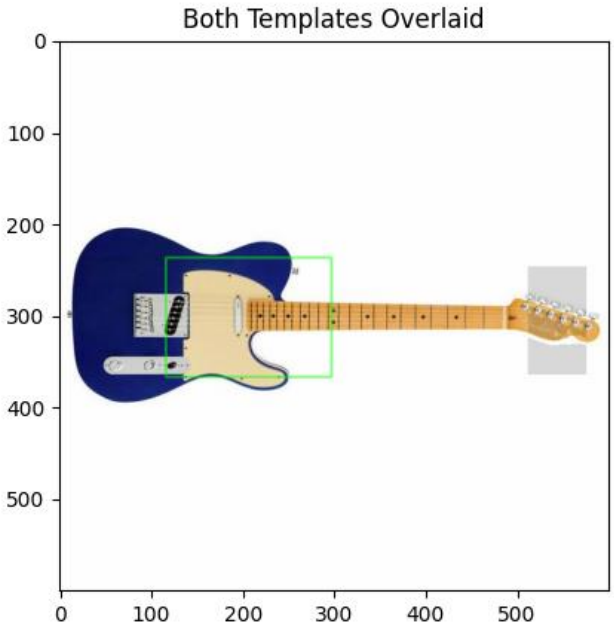


Figure 18: The output showing the best match for a specific template pair. The overlay indicates the matched regions with the highest cosine similarity score. The matched regions are clearly highlighted, showing the effectiveness of the algorithm in identifying the correct guitar body type. Both the Telecaster Head and Upper-Body Templates were Matched with the Largest Score

```
Checking match for tele_upper_body_template.png.npy and tele_head_template.png.npy...
Best rotation for Upper Body template: 90 degrees
Matching with template rotated by 0 degrees
Matching with template rotated by 45 degrees
Matching with template rotated by 90 degrees
Matching with template rotated by 135 degrees
Matching with template rotated by 180 degrees
Matching with template rotated by 225 degrees
Matching with template rotated by 270 degrees
Matching with template rotated by 315 degrees
Best rotation for Head template: 270 degrees
Checking match for strato_upper_body_template.png.npy and strato_head_template.png.npy...
Best rotation for Upper Body template: 90 degrees
Matching with template rotated by 0 degrees
Matching with template rotated by 45 degrees
Matching with template rotated by 90 degrees
Matching with template rotated by 135 degrees
Matching with template rotated by 180 degrees
Matching with template rotated by 225 degrees
Matching with template rotated by 270 degrees
Matching with template rotated by 315 degrees
Best rotation for Head template: 270 degrees
Checking match for jazz_upper_body_template.png.npy and jazz_head_template.png.npy...
Best rotation for Upper Body template: 270 degrees
Matching with template rotated by 0 degrees
Matching with template rotated by 45 degrees
Matching with template rotated by 90 degrees
Matching with template rotated by 135 degrees
Matching with template rotated by 180 degrees
Matching with template rotated by 225 degrees
Matching with template rotated by 270 degrees
Matching with template rotated by 315 degrees
Best rotation for Head template: 270 degrees
Best matched pair: tele_upper_body_template.png.npy and tele_head_template.png.npy
Best rotation for Upper Body template: 270 degrees
Best rotation for Head template: 270 degrees
```

Figure 19: Example of Match Checking Procedure in Dominant Gradient Orientation Template Matching

5.3 Haar Classifier – Sample Results

Due to the limited size of our training and testing datasets as well as the limited training stages (the training period only lasted for less than 90 minutes in total, over 15 training stages), our classifier only managed to learn to capture the most obvious guitar part, its neck, as shown in Figure 21. The features it managed to extract with a 24x24 window used were 261,600, and it managed to achieve a maximum false negative/alarm rate of 0.3, as shown in Figure 22.

Figure 20: Small Haar Classifier in Action – its Ability to Only Capture Guitar Necks was Noticeable

```
===== TRAINING 15-stage =====
<BEGIN
POS count : consumed   170 : 292
NEG count : acceptanceRatio    0 : 0
Required leaf false alarm rate achieved. Branch training terminated.
```

Figure 21: Message Output after Training the 15th Stage of the Haar Classifier

6. Conclusions

Our effort explored three distinct approaches to develop a tool for automatically detecting guitar body types in images. Each approach was evaluated based on its effectiveness, robustness, and potential for practical application.

6.1 Template Matching-Naïve Approach

The first approach used a naïve template-matching technique with whole-body binary masks. It was easy to implement but limited in effectiveness due to dependency on template quality and sensitivity to lighting, angles, and background clutter. It struggled in complex, real-world conditions, highlighting the need for more advanced methods.

6.2 Template Matching-Enhanced Approach

The second approach utilized dominant orientations and gradient analysis. This method improved over the naïve technique by focusing on local image gradients, making it more robust to variations in lighting, angles, and partial occlusions. It provided better detection accuracy and flexibility for different guitar shapes and sizes. However, it still faced challenges in complex scenes with significant illumination or background noise. Despite these issues, it showed higher accuracy and adaptability, making it a solid foundation for further development.

6.2 Haar Classifier Approach

The third approach trained a Haar classifier to distinguish guitar body types. It showed the highest accuracy, effectively handling variations in lighting, angles, and background clutter. Scalable and adaptable, it could improve with more labeled data and recognized new guitar models over time. The Haar classifier demonstrated the most promise, offering a robust and scalable solution.

Each approach contributed valuable insights. The naïve template-matching technique served as a starting point, the gradient-based method improved robustness and flexibility, and the Haar classifier approach offered high accuracy and scalability. These findings lay the groundwork for further research to enhance the tool’s effectiveness.

References

Bennett, J. (2021). *The guitar as a creative and cultural practice*. *IASPM Journal, 11*(2), 1–12. https://doi.org/10.5429/2079-3871(2021)v11i2.1291

Bradski, G., & Kaehler, A. (2000). *OpenCV library: Open source computer vision*. Retrieved December 10, 2024, from https://opencv.org

Cottonbro Studio. (n.d.). *Tennis racket and ball on clay court* [Photograph]. Pexels. Retrieved December 18, 2024 from https://www.pexels.com/search/tennis%20racket%20image/

Daftry, S., Ridge, B., Seto, W., Pham, T.-H., Ilhardt, P., Maggiolino, G., Van der Merwe, M., Brinkman, A., Mayo, J., Kulczycki, E., & Detry, R. (2021). *Machine vision based sample-tube localization for Mars Sample Return*. IEEE. https://doi.org/10.48550/arXiv.2103.09942

Han, C. (2023). *Fender's historical position in the electric guitar family*. *Frontiers in Art Research, 5*(17), 27-31. https://doi.org/10.25236/FAR.2023.051705

Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., & Lepetit, V. (2012). *Gradient response maps for real-time detection of texture-less objects*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(5), 876–888. https://doi.org/10.1109/TPAMI.2011.206

Microsoft. (n.d.). *Playwright: Fast and reliable end-to-end testing for modern web apps*. GitHub repository. Retrieved December 1, 2024, from https://github.com/microsoft/playwright

Mrnugget. (n.d.). *opencv-haar-classifier-training* [GitHub repository]. GitHub. Retrieved December 17, 2024, from https://github.com/mrnugget/opencv-haar-classifier-training

OpenCV. (n.d.). *Cascade classifier training*. OpenCV Documentation. Retrieved December 13, 2024, from https://docs.opencv.org/4.x/dc/d88/tutorial_traincascade.html

OpenCV. (n.d.). *Contours: Getting started*. OpenCV Documentation. Retrieved December 10, 2024, from https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html

OpenCV. (n.d). *ORB (Oriented FAST and Rotated BRIEF)*. OpenCV Documentation. Retrieved December 17, https://docs.opencv.org/4.x/d1/d89/tutorial_py_orb.html

Pelayo, J. M. G. III, Mallari, S. D. C., & Pelayo, J. J. S. (2015). *Guitar as the preferred musical instrument*. Center for Research and Development, Systems Plus College Foundation. U.S. Department of Education. Retrieved December 3, 2024, from https://files.eric.ed.gov/fulltext/ED560674.pdf

Python Software Foundation. (n.d.). *Python programming language*. Retrieved December 10, 2024, from https://www.python.org

Rayna, T., & Striukova, L. (2018). *Gibson vs. Fender: Innovation paths in the early electric guitar industry (1945–1984)*. In *XXVIIe Conférence Internationale de Management Stratégique*, Montpellier, France. Retrieved December 3, 2024, from https://www.strategie-aims.com/conferences/29-xxviie-conference-de-l-aims/communications/5079-gibson-vs-fender-innovation-paths-in-the-early-electric-guitar-industry-19451984/download

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *International Conference on Computer Vision*, 2564–2571. https://doi.org/10.1109/iccv.2011.6126544

Thomann GmbH. (n.d.). *Thomann online store*. Retrieved December 1, 2024, from https://www.thomann.de

Viola, P., & Jones, M. (2001). *Rapid object detection using a boosted cascade of simple features*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1, 511–518. https://doi.org/10.1109/CVPR.2001.990517