Detecting Hands in Renaissance Era Paintings

through a combination of multiple cues

Gerard J. Meier

g.j.meier@students.uu.nl / gerjoo@gmail.com

Master thesis
ICA-4193369
Universiteit Utrecht
July 6, 2018

Contents

Chapter 1

Introduction

Hand recognition in images is a broadly researched subject. The subject can be divided into two main areas of research, finding hand positions and estimating a hand's pose. Pose estimation finds use in (e.g.) human computer interaction (HCI) [61] by for example, allowing a user to navigate through a virtual world using gestures, as well as, automated sign-language transcribing [37]. Hand position determination will work well in conjunction with the former in a single pipeline, e.g., once a hand position is found a pose estimator can uncover specific hand gestures and hand orientations. These two uses are typically researched separately. This thesis will focus on determining the position of hands. Moreover, this thesis will not focus on typical hand detection in photos of humans [36, 47, 32, 43], but rather in Renaissance era paintings.

Renaissance era paintings, or paintings in general, are a challenging subject because they are an artistic representation of reality. This, at times, results in contrived poses and complicated interactions between subjects. Also the use of colors may vary heavily, based on composition, availability of dyes and simply decay over time. The focus is placed on the Renaissance era because (1) it includes many reasonably realistic paintings, and (2) it is a well known era making it an interesting area to study.

This thesis sets out to detect hands in Renaissance era paintings using a combination of object detectors, low-level cues supported by a support vector machine and statistical priors that describe hand sizes and positions. The object detectors include Viola and Jones [56] framework for object detection and a convolutional neural network based human pose detector by Insafutdinov et al. [25]. The low-level cues will focus on color and shape. It will be shown that a combination of these multiple components will lead to better hand detection performance over the use of a single component.

Closely related to this thesis is the work by Westlake et al. [57] on detecting persons in paintings, some of their paintings are from the Renaissance era. Their work varies by only detecting persons, rather than hands. Their work does highlight the challenge of detecting objects in paintings. Similar to Westlake et al. [57] is the work by Ginosar et al. [16], they focus on detecting people in surrealistic Cubism artwork. Schlecht et al. [45] present work that detects hand gestures in medieval books, although similar to this thesis, their work is limited to just a handful of gestures, with a dataset that has little variance in terms of color and composition.

Contributions of this thesis are:

1. The introduction of a hand and head annotated dataset as used in this thesis, and for future work in the field[1];
2. The first ever benchmark for hand detection in Renaissance era paintings;
3. A low-level description of colors and spatial priors in Renaissance era paintings;
4. Evaluation of the applicability of several low-level cues and frameworks for detecting hands in Renaissance era paintings;
5. Evaluation of the usefulness of combining multiple cues to detect hands.

This thesis is structured as follows: A review of related work is presented in Section 2. Following that in Section 3 a discussion of the dataset and its characteristics. Section 4 details the pipeline and its components used to recognize hands and motivates the design choices thereof. In Section 5 the proposed pipeline is evaluated for its effectiveness by optimizing individual components leading to an overal improved performance, this section also highlights the strengths and weaknesses of the pipeline. The final section concludes this thesis and discusses opportunities for future work.

[1] Paintings and annotations downloadable at `https://github.com/gerjo/paintings`

Chapter 2

Related Work

This chapter surveys existing work pertaining to hand recognition. The majority of techniques in this section are already successfully applied to the task of hand recognizing hands in photos of humans, however, they are not technically limited to just such photos. The first sections discuss techniques using different cues: color, shape and hand-context. The final section covers some overarching work which discusses the applicability of classifiers trained for object recognition in photos applied to paintings.

2.1 Hand Color Based Detection

A great cue towards detecting hands is the ability to discover skin regions. Hand detection using color has been approached in two ways: thresholding and distribution models. Both techniques follow the assumption that skin color can be quantized and generalized across multiple images.

Thresholding is based on a rule-based system. Color ranges representing skin are formulated using prior information. For example, [33] describe the color of each pixel using separate RGB values and assign a hand/non-hand label to the pixel depending on whether individual RGB components are within a threshold. When neighboring pixels carry the hand-label they are associated to the same skin-region. These regions then describe the location of bare body parts. In their system a shape criterion is used to determine whether the shape of a region matching that of a face. It stands to reason that a shape criterion describing hands could be formulated. Gomez and Morales [17] present a method to automatically generate such thresholding rules. Their thresholding rules are elaborate with (e.g.,) addition and multiplication of color components prior to subjecting them to a threshold.

The other key technique is based on distribution models. Prior color information is used to construct a global probability distribution of whether a given pixel is skin-colored [29]. To detect skin, local distributions are generated and compared for similarity with the global model. One way of obtaining local distributions is through the sliding window approach [21]. This approach subdivides the input image into fixed-size windows. Notable parameters include the window size and the offset between each window's center. The offset i.e. step-size could be left at a single pixel for an exhaustive result. A key assumption is the window size, which must closely match the size of the to be detected object. Some implementations circumvent the fixed window size assumption by using a so-called image pyramid [56]. With the image pyramid the window size is fixed, but the input image is downscaled until a minimum size is reached.

For distribution models Phung et al. [42] found that the performance remains largely unaffected with respect to the used color space, as long as both chrominance and luminance channels are used in the distribution model. The latter implies that just looking at single channel, such as Hue from the HSV [55] color space, is not sufficient.

Instead of fully relying on a global color model applicable to all input images, it is possible to construct a global model specific for an single image. Hsieh et al. [23], Fritsch et al. [15] generate such models at run-time through automated face detection. When a face is detected, skin pixels are extracted to generate an image specific model. These models are more robust to local variations in lighting, following from the assumption that light conditions within a photo remain reasonably consistent. A variant on this approach is used by Taylor and Morris [51], skin color models are gathered automatically by running a face detection algorithm configured to run in a high-precision moderate-recall configuration. This allows them to bypass the manual annotation step and easily apply their system to new image sources.

Whether using thresholding or distribution models, the resulting skin regions still need to be classified which type of object they are, if any. Phung et al. [41] propose the use of edge detection to break detected skin regions into multiple segments by detecting contours, this also results in a separation between false-positive backgrounds and true-positive objects. Afterward s they apply a thresholding algorithm to discart small regions and assume the remaining pieces are body parts. Mittal et al. [36] fit a straight line segment to the resulting skin regions and propose that either end of the line contains a hand; this follows from the assumption that a part of the arm is visible, e.g., when wearing a t-shirt. Their system discards regions representing faces by applying face detection. Panin et al. [39] apply a similar fitting approach, at the core of their algorithm circles are fitted to regions, the largest circle is assumed to be the hand palm. Their work is limited to images containing just one hand captured from a top-down view point. The added benefit of fitting a circle is that the scale and palm of the hand is known, and subsequent algorithms can take this into account when determining a hand's pose.

2.2 Hand Shape Template Matching

Hands may be found by searching for their distinctive shape across an image. A typical approach is template matching; which relies on a pre-created database with hand templates i.e., examples. During the detection phase, tentative hands are compared against these templates. When a similar template is found, the tentative hand is considered an actual hand, supplemented with a likelihood as given by the similarity measure.

A frequently used technique for encoding templates is the use of edges. Edge detection is typically done by convolving the input image with a convolution kernel. These kernels are discrete approximations of oscillating functions such as Laplacian of Gaussian [19] or are a measure of directional local differences such as the Prewitt operator [10]. Edge detection can contain false positives, i.e., an edge is found where there ought to be none. This could occur when the input image contains high-frequency noise, or under varying light conditions such as specular highlights. A typical approach is to blur the image beforehand (variance reduction) and after edge detection to apply non-maximum suppression to thin the redundant edges, as documented by Canny [5]. Jesorsky et al. [27] use the Hausdorff distance metric to compare the edge detection results with a database of templates. Hausdorff distance associates each input pixel in an edge to the nearest pixel in the template's edge; the resulting similarity measure is the distance between the farthest associated pixels. Hausdorff distance heavily penalizes a single mismatch among the edges. Athitsos and Sclaroff [1] overcome this by using chamfer distance, which rather than taking the maximum distance, uses the average of all distances. Veltkamp and Hagedoorn [54] provide a survey of matching techniques.

Edge detection based template matching for hands is tricky due to the high degree of deformity, this calls for either a large database of templates or a distance metric which is invariant to hand pose orientations. Existing work typically limits itself to the detection of a few hand poses. Schlecht et al. [45] detect hands in medieval book drawings using just 10 templates. The latter requires few templates because their domain of application contains only a few poses with minimal variance. The high degree of deformity is highlighted by Athitsos and Sclaroff [1], in their work they use 26 poses where each pose is captured from 86 viewpoints using 48 different rotations yielding a total of 107,328 templates. Their templates are automatically generated by using 3D renders of an articulated hand. Stenger et al. [47] also generate the templates from a 3D model, but extend on the work by constructing a hierarchy such that the search space can be pruned.

Another technique to encode templates are shape context descriptors [3]. These descriptors still rely on the output of edge detection, but rather than store the cartesian coordinates of a pixel, store the quantized distances and angles between all edge pixels in an image. The resulting descriptor can be quite large, the authors recommend uniformly sampling the results of edge detection. Key advantages of shape context descriptors are invariance to rotation and limited invariance to scale due to quantization of distances [52]. Ong and Bowden [38] successfully use shape context discriptors to cluster similar looking hands. It must be noted that in their work the descriptors are not evaluated for false-positives nor false-negatives.

Zondag et al. [65] use histograms of oriented gradients (HOG) to encode hand poses. HOG quantises the image into a grid and per cell creates a histogram of the direction of gradients. The direction is computed by treating the outcome of two separate edge detection filters (e.g., [10]) as a single vector. HOG is typically employed in conjunction with a support vector machine (SVM) [4]. In essence, the SVM aggregates the examples database used in earlier discussed methods. HOG allows some degree of variance due to quantization in the histogram, but still relies on the SVM to be trained with sufficient examples.

Template matching works well in conjunction with color matching techniques, as demonstrated by Stenger [48]. In their work, skin color is used to find tentative hands. When skin color is not available or reliable, an exhaustive sliding window approach can be used to systematically scan the input image for hands, as done by Schlecht et al. [45].

Not all techniques rely on encoding a single hand as a template. Viola and Jones [56] introduce a framework that relies on small local contrast descriptors taken at various resolutions, which when combined describe a shape. Using a boosting algorithm, thousands of these small descriptors are placed in a tree hierarchy which can be evaluated against the to be detected object. Zondag et al. [65] show that this approach can work well for hand detection, but limit their dataset to images containing a single open hand with unconstrained fingers. Kolsch and Turk [31] evaluate the rotational invariance of this framework for detecting hands. Their findings indicate the framework's performance to quickly deteriorate under rotation. Furthermore, they train a classifier per hand orientation, with their dataset limited to just 6 different poses. Ong and Bowden [38] present a similar framework, with an emphasis on hand detection. Their framework varies from Viola and Jones [56] by including a clustering algorithm to account for the high deformity of hands, and so build multiple trees. They evaluate their framework, with a high success rate, using images taken from the sign language domain which carry a simple background. Ong and Bowden [38] indicate in their conclusion that future work should focus on improving the performance on a cluttered background. The number of training images required makes this type of framework impractical for renaissance paintings, for example Viola and Jones [56] use nearly 5000 faces in their original face detection research to create a single classifier. In later work, Kölsch and Turk [32], provide an a priori estimator to determine the classification potential of training data. This however just reduces the training time by culling the input data, the number of high quality annotations required remains the same.

The choice of how to encode a template is important. A very generic template may conveniently describe several hand poses, but will proportionally also yield false positives. Conversely, a very specific template may avoid false-positives, but requires a fast database of many such templates.

2.3 Hand Context Modelling

A cue towards a hand's position may be found by looking at the context in which a hand resides. This could simply be the arm and by extension the torso, or on a finer scale, a hand would be characterized by the presence of a palm and fingers. There may also be latent background traits that characterize the presence of a hand, for example, a painter may prefer to painting a hand on a contrasting background to aid composition.

To explicitly capture the relation between different features such as body parts, part-based models are used. Felzenszwalb et al. [14] presents such a system to detect humans. Their system has two layers. At the top layer sits a single HOG descriptor which coarsely describes the entire object. At the lower layer sit multiple fine-grained HOG descriptors describing individual parts, such as the phalanges of a hand. During the detection phase, a likelihood map for the coarse descriptor is generated; in a second phase, the system searches for the finer descriptors positioned relative to the coarse descriptor's most likely location. This approach allows a degree of positional freedom among the different descriptors. Mittal et al. [36] demonstrate the use of Felzenszwalb et al. [14] system for detection of hands. They use two models, one describing the hand itself, and one including the hand's context which is the area around and including the wrist.

Not always does the context have to be modelled as explicitly. Dardas and Georganas [9] use SIFT [34] to automatically detect keypoints which describe the hand and surrounding background context. These SIFT keypoints are characterizing areas which remain visible as the image is resized to a lower resolution. At each keypoint a local descriptor is generated, such as a directed color gradient. At the detection phase, SIFT is again used to generate descriptors, which are then compared to the database of previously created descriptors. More broadly, this is an application of the so-called bag-of-words (BoW) model. With BoW multiple local descriptors (e.g., one descriptor per keypoint) are associated with and object (hand or hand-pose), if during the detection phase enough of those descriptors are also present, it's assumed to be match. A key shortcoming of BoW is that the spatial relation between local descriptors is ignored, simply a sufficient number of similar descriptors is considered enough to indicate whether something is a hand.

It is possible to model the relation between an object and human pose. Yao and Fei-Fei [62] demonstrate such system by modelling the position of sports equipment and a sportsperson. Their system relies on prior information to capture the spatial relation, for example a tennis racket is most often found near a hand. This approach will only work when a common object such as sports equipment or music instruments are available in the image. Bambach et al. [2] take advantage of prior knowledge about a hand's likely location within an image. Their system works with footage taken from a body-mounted camera, where the to be detected hands belong to the camera's wearer. Initial hand proposals in their system are selected with a bias towards proposals near a common hand location.

Insafutdinov et al. [25] presents a state-of-the-art human pose detector based on convolutional neural networks (CNN) [18]. A CNN is used to individually find body parts, inclusive of hands. After tentative body parts are found, a separate algorithm connects the body parts to form a person. This uses mutually exclusive posterior information such that multiple persons can be found, even when body parts are visually occluded. This two-step system is quite computationally intensive.

Savalle et al. [44] present work that introduces DPM semantics into CNNs, that is, introduces a notion of spacial relation between features. Their work also indicates that a CNNs will nearly always outperform HOG based DPMs as presented by Felzenszwalb et al. [14].

2.4 Hands Depicted Across Different Domains

The application of object detection algorithms designed for natural photos on paintings is not uncommon. Hall et al. [22] describe this as the cross-depiction problem, in which knowledge (e.g., skin color distributions) learned in one domain is transfered and applied to another (i.e., transfer learning). Their key findings are that (1) appearance-based recognition systems tend to be overfitted to one depiction and (2) models that explicitly encode spatial relations between parts are more robust.

Wu et al. [60] use a part-based model to detect objects, inclusive of persons, in paintings. The individual part-models are learned from photographs and then applied to a wide range of styles, such as children's drawings and realistic paintings. Their system uses the DPM presented by Felzenszwalb et al. [14], but with modifications to account for overfitting on photographs: there is no root descriptor that coarsely describes the entire object and the fine grained part descriptors are no longer applied mutually exclusively. In earlier work, Wu and Hall [59] successfully overcame the overfitting issue of a DPM by describing parts using primitive shapes (e.g., rectangles and circles) instead of histograms of oriented gradients.

Westlake et al. [57] present work that focusses on using CNNs for people detection in paintings. Their CNN is trained using natural photos and then applied for people detection in various painting styles such as High Renaissance and pop art. The overfitting issue is overcome by tweaking the CNN parameters to increase performance on a painting-only validation set. This is done by training the CNN using photos, after the network converges to a solution, certain layers are fixed, and then training is resumed but this time using a painting dataset. With these tweaks, they manage to improve the Average Precision (AP) of the best performing configuration from 43% to 58%. Yosinski et al. [63] discusses this process as a form of transfer learning. This is particularly relevant when the available annotated dataset is small, such as with Renaissance era paintings. CNNs have an internal hierarchy of layers. Low-level layers capture features such as edges, whereas high-level layers capture abstractions such as shapes. Yosinski et al. [63] indicate that low-level layers are overfitted to the input data's domain, and discuss how these can be retrained to fit data from another domain, while keeping the high-level layers fixed; which is exactly what Westlake et al. [57] have done in their work.

Chapter 3

Dataset

This section introduces the painting dataset as used in this thesis. First a discussion of existing datasets is presented. Following that we delve deeper into the characteristics of the used datasets. This also highlights some of the challenges that are present with detecting hands in Renaissance era paintings. The first three sections discusses the source of the dataset, annotation process and some meta data. The final sections examine statistical properties with respect to color and spatial positions of hands.

3.1 Existing datasets

One requirement of the painting dataset is the availability of annotated hand locations. There are datasets with annotated hands available [11, 36], but these pertain to just natural photos. Westlake et al. [57] present an artwork dataset inclusive of Renaissance era paintings, but is limited to annotations of people in the form of a bounding rectangle. This thesis will use the paintings from WikiArt [58] as retrieved late 2015, this is the same source as Westlake et al. [57] used.

3.2 Painting Categories

A focus is placed on four different painting categories: *Tenebrism*, *Academicism*, *Early Renaissance* and *High Renaissance*. These categories are selected for their availability of meta annotations and because works are available from a mixture of artists. This set also limits the scope of to work to just realistic representations of humans, unlike e.g., *Renaissance Surrealism.*

There may be some semantic overlap between the categories, such as High Renaissance being a time period rather than a style such as Academicism, and an artist may adhere to Academicism while also painting with a

Tenebrism style. These category annotations are provided mutually exclusive by WikiArt [58], and will be used verbatim.

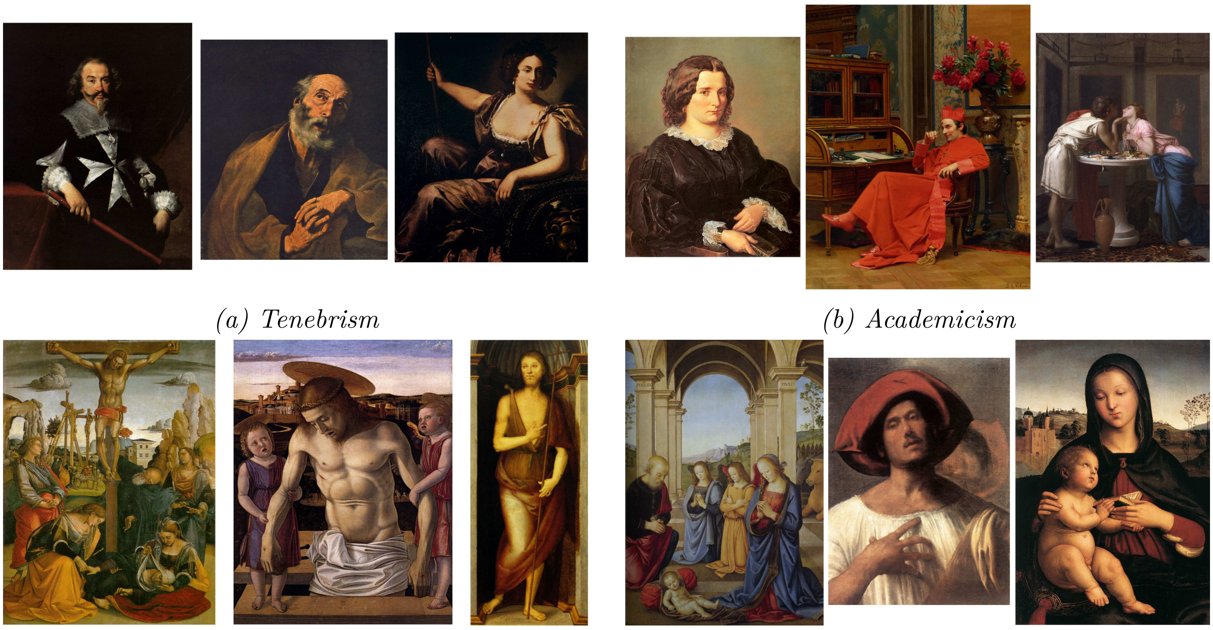
Figure 3.1 shows typical examples for each category as used in the dataset. A brief description of the selected categories is as follows [7];

Tenebrism (17th century) - an art style that is recognized by depicting the painting's subject in light colors, on a very dark background. The subject appears to emerge out of the darkness, with usually just a single source of light.

Academicism (16th-20th century) - a movement that adheres rigorous training and atomically correct artwork, paintings typically convey an intellectual topic in an idealistic setting.

Early Renaissance (1400 to 1490) - refers to works created during the early renaissance years. Artists started to focus on anatomically accurate paintings and the notion of vanishing points took existence.

High Renaissance (1490 to 1527) - the follow up period of the latter. A notable difference is the added realism in most faces, which during the Early Renaissance was reserved just for religious characters, such as angels and depictions of Jesus Christ.



(a) Tenebrism

(b) Academicism

(c) Early Renaissance

(d) High Renaissance

Figure 3.1: Example paintings from WikiArt [58] as used used in this thesis.

3.3 Annotation Process and Filtering

Not all paintings are considered suitable for annotation. Several paintings were excluded for having more than five persons, no persons at all, or having tiny hands with either dimension smaller than ten pixels. Westlake et al. [57] also exclude some of the WikiArt paintings based on the contents being annotated as “difficult”, no furter details are provided. Table 3.2 lists the number of persons annotated per painting. The raw WikiArt dataset also includes pictures of statues, stained glass and charcoal drawings, these are filtered as well. Non-rectangular paintings are kept. Figure 3.3 shows some typical excluded work. Most of the excluded work comes from the Early and Heigh Renaissance categories; the source material contains more low resolution images leading to frequent small hands, as well as depictions of biblical scenes with many persons present (e.g., crucifixion of Jesus Christ). This is in contrast to the Tenebrism category, which usually pertains a specific subject with few bystanders. Table 3.1 shows these statistics, along with the number of hands and faces annotated per category.

Annotations are created using a tool specifically designed for this task. Figure A.3 shows a screenshot of the tool in use. The tool allows placement and modification of rectangles i.e., annotations. Individual annotations are associated with a person through a context sensitive menu. A zoom feature is available to allow accurate annotation of smaller objects. The resulting rectangle is set to encompass all fingers and cover the hand palm, or cover the majority of the face. Figures A.1 and A.2 show examples of annotated hands and faces. Faces are also annotated to allow a greater range of experiments to take place as part of this thesis. Extra care is taken to fit the annotation rectangle snugly around the object to include as much skin color as possible while reducing background colors.

The remaining sections in this chapter will statistically examine the annotated data in the dataset.

Category	**Downl.**	**Annotated**	**Excluded**	**Artists**	**Faces**	**Left**	**Right**
Tenebrism	201	105	23	10	210	179	162
Academicism	295	100	42	24	158	143	134
Early Ren.	318	104	131	28	242	213	205
High Ren.	298	100	68	20	213	180	169

Table 3.1: Statistics for the dataset. The downloads column refers to the number of paintings available at the time (early 2015).

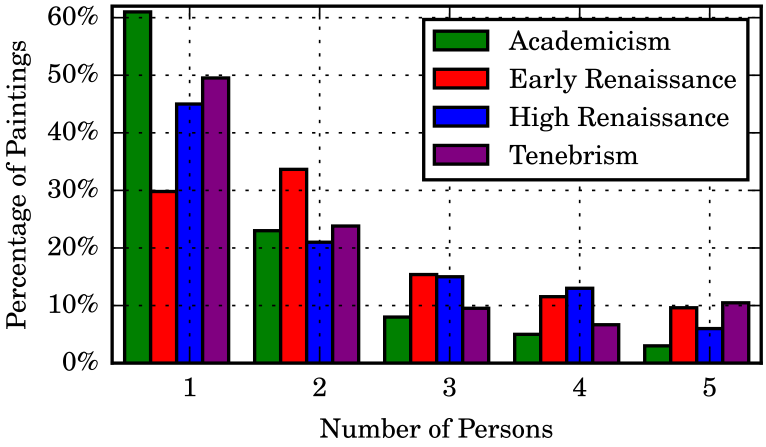
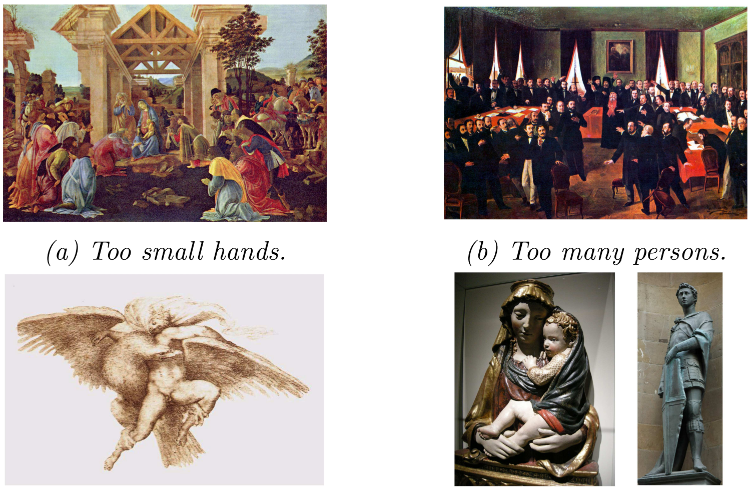


Figure 3.2: Number of persons per painting in the annotated dataset. Paintings with more than five people are typically excluded due to having too many small hands.



(a) Too small hands.

(b) Too many persons.

(c) Charcoal sketches.

(d) Statues and ornaments.

Figure 3.3: Typical excluded images. There is a strong occurence correlation between too many people and small hands.

3.4 Color Statistics

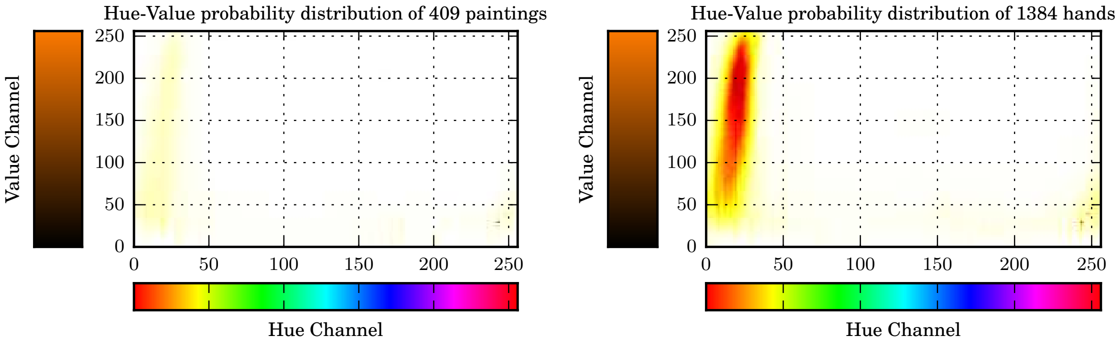
A typical way to describe colors is through the *hue, saturation and value* (HSV) color space [46, 55]. The hue channel represents the predominant absolute color, saturation describes the strength of the hue channel, whereas value indicates the darkness. The motivation behind HSV is that it introduces a certain level of light intensity invariance within the hue channel. There are similar trend, such as YCrCb and CIELab, each with the same purpose of decomposing a color into chrominance and luminance/intensity channels.

To gain insight into the colors used in all paintings refer to Figure 3.5a. Adjacent is Figure 3.5b which shows the use of colors in the annotated hands. Both plots are made using [26] implementation of HSV. For graphing convenience the hue channel's visual range is upscaled from $[0, 180)$ to $[0, 256)$ to match the other channels. To assure unbiasedness towards large paintings when aggregating results, all input images are resized to the same resolution using bilinear interpolation. The number of bins is 256 which exactly matches the number of unique values per channel. The individual channels are normalized such that the sum of all bins combined equals 1, i.e., it represents the probability rather than absolute quantity.

It is interesting to observe that the colors in both histograms appear to peak with a high probability around the 'amber' hue. This might be due to the aging of certain components in the dye. The former isn't as strongly with a dataset containing humans; figure 3.6 shows the HSV histogram from a photo dataset [36], which has a strong presence of the 'blue' hue. This photo dataset also uses rotated bounding boxes to annotate hands, which implies there is less background color in the plot. To understand the correlation between the hue and value channels, refer to Figure 3.4. It is observed that hands peak very strongly around a value of 200. These plots motivate the need to look at more than just the hue channel. Hue and saturation are distributed in a similar fashion, and also show a strong peak at a particular saturation level.

Figure 3.7 shows the hue and value plot of each category. Early and High Renaissance categories are similarly distributed, which is to be expected due to the successive time period and the same artists contributing to both genres. Tenebrism shows more darker colors than the others, this is in tune with the characteristics that category. The Academicism plot has a narrower peak, this may be attributed to the standardized norms for painting techniques and colors set forth by an academy [7]. As with the aggregated HSV plots of all paintings (Figure 3.5), there is a noticeable lack of 'blue' in the hue channels.

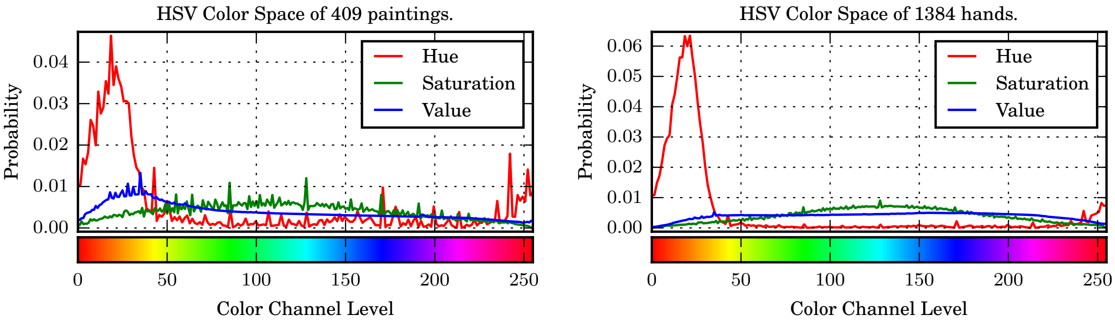
Color-based techniques can struggle to generalize across different ethnic groups [29]. Through manual observation it appears that the dataset contains no black people. This appears to be a trend with typical renaissance artwork.



(a) Annotated paintings

(b) Annotated hands

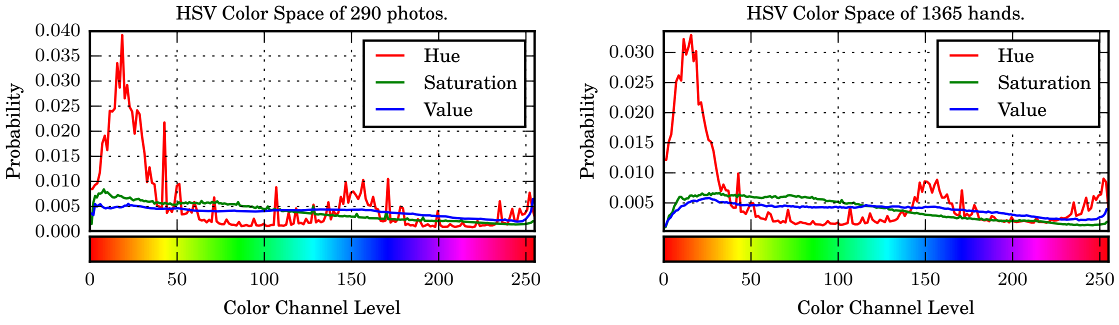
Figure 3.4: Hue and value pairwise probability plot. For visual convenience a hue bar is included along the x-axis. The y-axis contains a value bar for the 'amber' hue with saturation set to maximum. Figure A.7 repeats the figures for photos. Probability runs from red (high) to yellow (medium) to white (never).



(a) Annotated paintings

(b) Annotated hands

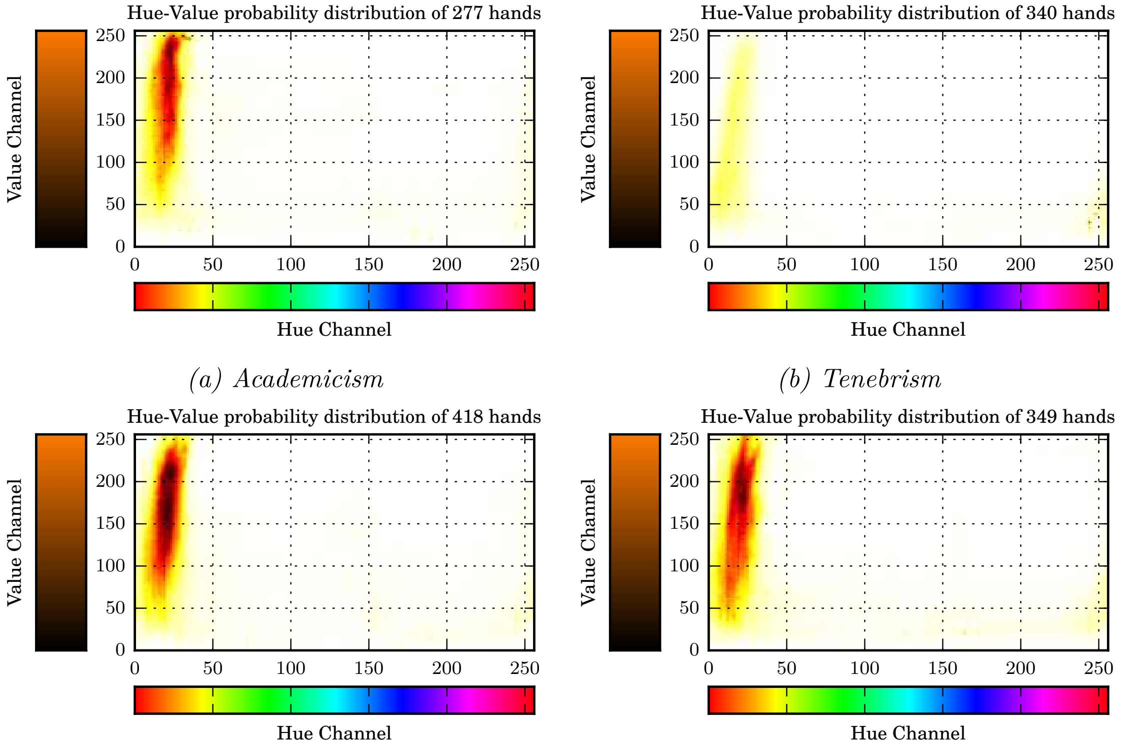
Figure 3.5: HSV color space histogram, the channels are plotted independently. For visual convenience a hue bar is included along the x-axis.



(a) HSV in photos

(b) HSV in real hands

Figure 3.6: HSV color space histogram of photos. This is a plot of the dataset as used by Mittal et al. [36] and consists of persons in assorted environments.



(a) Academicism

(b) Tenebrism

(c) Early Renaissance

(d) High Renaissance

Figure 3.7: Hue and value pairwise probability plot per painting category. For visual convenience a hue bar is included along the x-axis. The y-axis contains a value bar for the 'amber' hue with saturation set to maximum. Probability runs from red (high) to yellow (medium) to white (never).

3.5 Skin Color Self Similarity

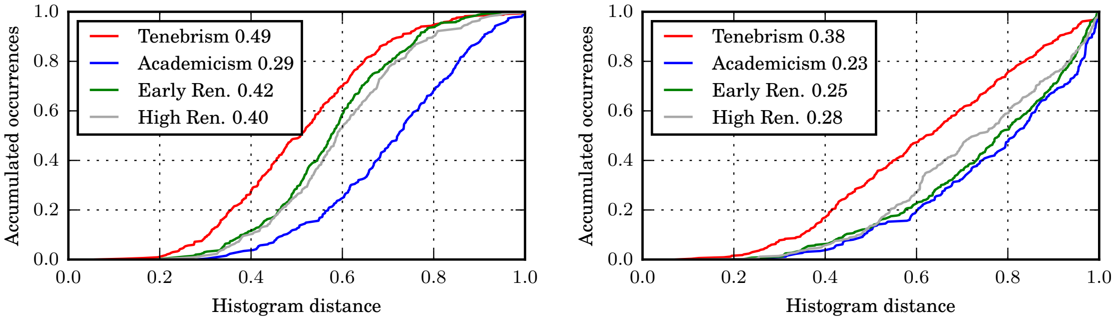
The previous section examined colors in hands and paintings. The resulting graph showed that there is some difference between the two. In particular hands show a strong peak around the ‘amber’ hue, whereas with the whole painting this peak is less strong.

Comparing the two graphs is informative, but in this case not necessarily indicative of the discriminatory factor of colors within a single painting. To determine whether hand colors are unique from other colors in a painting a comparison experiment is set up. For each painting the similarity is measured between colors from an annotated hand and colors in the entire painting. Ideally the colors of a hand would be unique, such that color could directly be used as a descriptor to distinguish between hand and non-hand. For comparison the experiment is repeated using random hand-sized patches from the painting. It is expected that random patches more closely resemble the colors in a painting than hands would. The colors are aggregated in a normalized histogram using the HSV color space and 32 bins per channel such that the total bin quantity is $32,768$. Similarity is measured using Chi-Square distance. This metric reduces the dissimilarity when large numbers differ only a small amount by scaling the difference proportionally to the sum of each histogram bin (Equation 3.1).

$$\text{Chi-Square}(A, B) = \sum_{i}^{N} \frac{(A_i - B_i)^2}{A_i + B_i} \tag{3.1}$$

Figure 3.8 shows the result of the experiment. The graph represents a cumulative distribution function (CDF) of all distances in ascending order, e.g., 100% of the distances is at least 1 (very dissimilar) and conversely 0% of the distances is 0 (identical). The area under the resulting curve is shown in the legend, a higher number indicates more similarity (i.e., the curve reaches its maximum quickly). The results show there is a different trend among the various painting categories. Tenebrism hand color is the most similar the painting’s colors whereas Academicism is the most dissimilar. For comparison, the test is repeated but instead of using hands it uses random hand sized extracts from the paintings. The same trend persists, but in general the area under the curve shrinks, suggesting that random matches are more dissimilar to the painting’s overall colors than hands are. The CDF of similarity measures between paintings and hands appears to be sigmoid shaped, which is indicative of an underlying normal distribution for hand-colors, as this sigmoid shape is much less apparent with random extracts. Colors of random extracts are expected to be distributed much flatter, i.e., show negative kurtosis.

The work of (e.g.) Hsieh et al. [23], Fritsch et al. [15] uses adaptive skin color models. At run-time faces are detected, from which a color model is

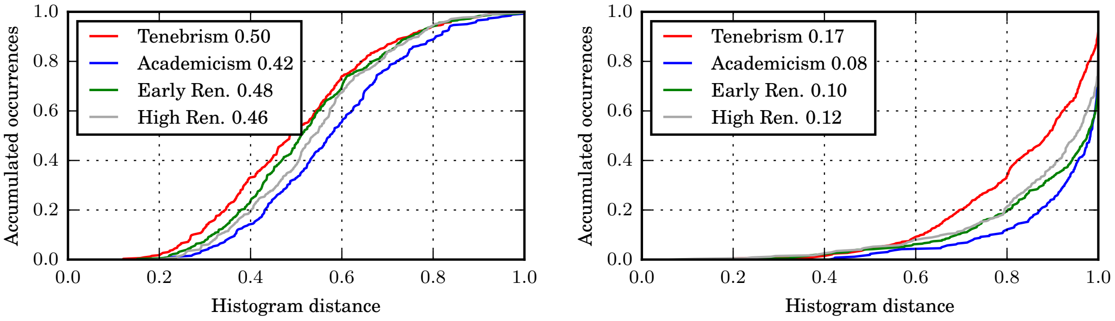


(a) Between paintings and hands

(b) Between paintings and random

Figure 3.8: Similarity between each painting and its hands or random patches. Occurrences are accumulated (CDF). A distance of 1 *indicates that histograms are very dissimilar,* 0 *indicates identicality.*

constructed and used to detect skin regions, which include hands. To determine the applicability of this approach, the above experiment is repeated with a variation. Rather than comparing hands to the painting, the hands of each person is compared to the person's face. Figure 3.9 shows the resulting CDF. Academicism shows the largest dissimilarity and Tenebrism the strongest similarity. This test is repeated by comparing faces with random patches. The results in Figure 3.9b indicate that random patches are vastly more dissimilar, thus proving that colors from faces could be used as a cue to detect hands.



(a) Between faces and hands

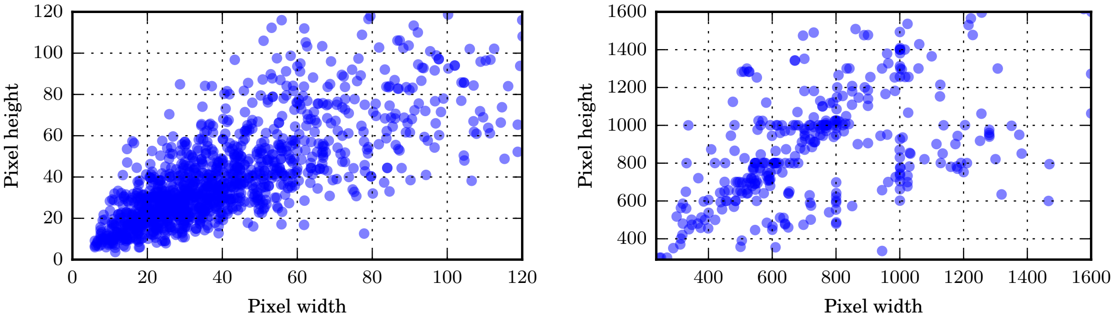
(b) Between faces and random

Figure 3.9: Similarity between face color and other parts of the painting. Occurrences are accumulated (CDF). A distance of 1 *indicates that histograms are very dissimilar,* 0 *indicates identicality.*

3.6 Hand Sizes

It's important the understand the size of hands in the dataset. If hands are too small, it may prove difficult to (e.g.,) generate color models due to shortage of color information. Hands in paintings have a mean of 90 by 87 pixels. Academicism is the category with the smallest hand size of 58 by 52 pixels. In related hand detection work, e.g., by Mittal et al. [36] hands have a mean of 30 by 43 pixels. This suggests that hand size should not necessarily be a limiting bottleneck.

The dataset contains paintings of varying size. It is common to normalize painting size, which will also reduce the variance of hand size between paintings. A great advantage of resizing in this case is the reduced memory footprint and increased processing speed of detection algorithms. To match the hand size of related work, it was decided to resize paintings such that no dimension exceeds 600 pixels while maintaining aspect ratios. This brings the mean painting size to 564 by 471 pixels and the mean hand size to 45 by 43 pixels. For reference, the size of photos in Mittal et al. [36] dataset is 402 by 459 pixels, ergo, hands in paintings are smaller, but this can be compensated for by using larger input images. Figure 3.10 shows the hand and painting sizes after normalization.



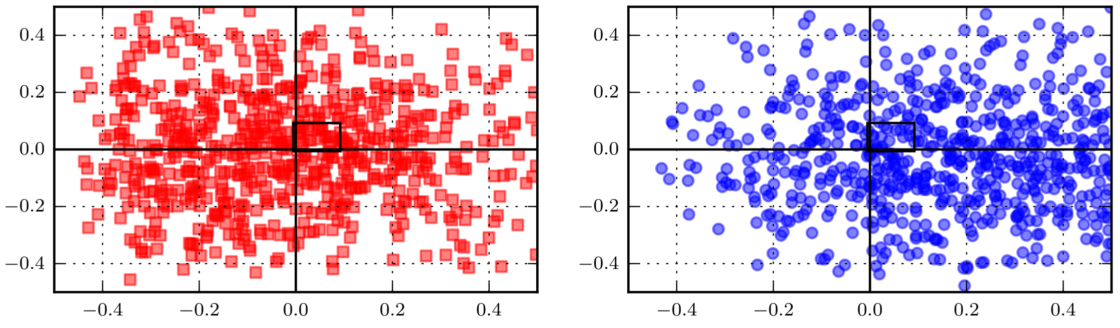
(a) Hand sizes.

(b) Painting sizes.

Figure 3.10: Hand sizes and painting sizes from the annotated dataset.

3.7 Hand Positions

Hand positions in paintings can be captured with a distribution. A study of the hand position shows them to be reasonably well spread on average across the painting. There does appear to be a bias for a person's left hand to be on the right hand side of the painting, as evidenced by Figure 3.11. This plot is computed by normalizing the hand's position with respect to a painting's size. Further analysis shows this trend to exist with all painting categories covered, see Figure A.6 for scatterplots per category.

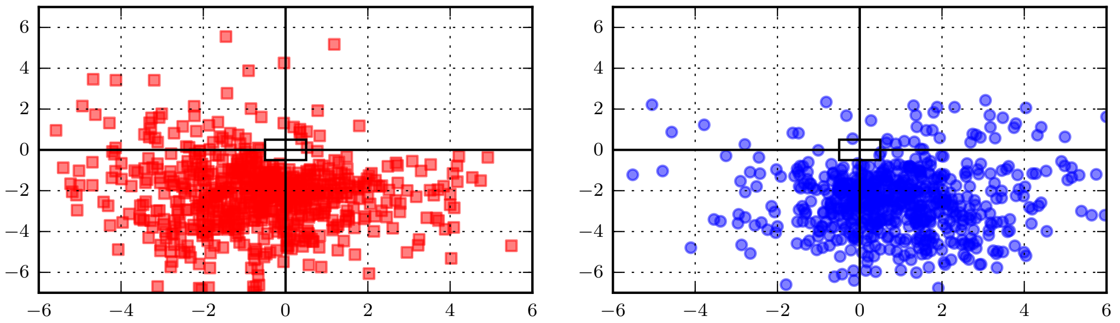


(a) *Locations of a person's right hand*

(b) *Locations of a person's left hand*

Figure 3.11: Absolute hand positions normalized by painting size. The axes indicate the painting's normalized size, with the paintings center in the origin. Results from all painting categories are combined. The square indicates the average face size and position.

It is also posible to describe a hand's position with respect to the person's head. To aggregate these results the face's center position is subtracted from the hand, and is divided by the face's size. The latter is required to account for paintings with a single person, which typically results in larger faces. In effect it describes the hand position in terms of face size. Figure 3.12 shows the resulting scatterplot. Left hands are on the left hand side of a person, and right hands conversely so. It also shows that hands are more likely to be below the head, as well as closer rather than farther away.



(a) *Relative right hands.*

(b) *Relative left hands.*

Figure 3.12: Positions of hands with respect to face size and position. The plot's origin indicates the center of annotated faces. The square indicates the average face size.

Chapter 4

Approach

The pipeline in this thesis uses a combination of multiple cues to detect hands. The cues are based on low-level image features such as color and edges, statistical priors of known hand locations and sizes, and hands as found by a state-of-the-art CNN human pose detector. Figure 4.1 shows a diagram of how the multiple cues interact and what their position is in the pipeline. The structure of the pipeline follows work from e.g., [36, 2, 48, 50] where multiple cues, such as skin color and HOG descriptors are used. The pipeline described in this chapter varies by generating initial proposals using Selective Search [53], and employs the multiple cues as a means to classify each proposal. In recent work Roy et al. [43] follow a similar approach to the pipeline in this thesis. They use a CNN to proces Selective Search proposals to detect hands in photos. An additional CNN trained to detect skin pixels attempts to reduce the number of false positives generated by the first CNN.

The structure of this chapter is as follows. The first sections discuss how initial hand proposals are generated. Following that two sections which discuss the processing and classification of low-level features, Section 4.4 introduces the DeeperCut CNN and the spatial priors cue is discussed in Section 4.5. The final section explains how the multiple cues are combined such that hands may be found.

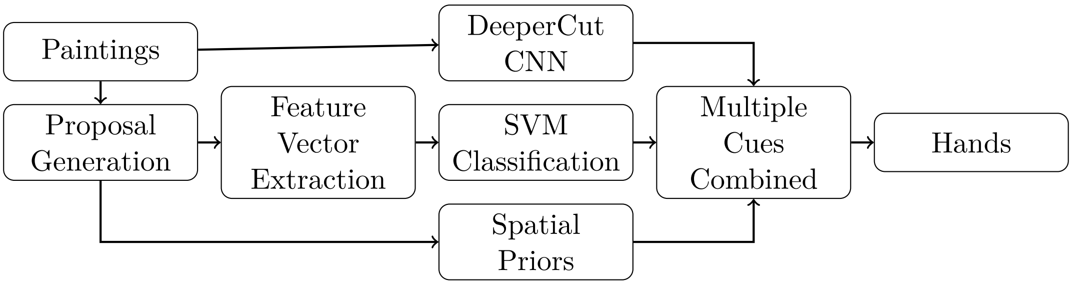


Figure 4.1: The proposed processing pipeline for detecting hands in Renaissance era paintings.

4.1 Tentative Hand Proposals

The first step in the pipeline is the generation of tentative hand proposals. A tentative hand proposal is a small region of the input image which might contain a hand. In its simplest form proposals can be generated using a sliding window [21]. A sliding window is a fixed size window moved pixel-by-pixel across the input image. Pixels inside the overlapping window as it slides are considered a hand proposal. The sliding phase is often repeated with varying window sizes across the same input image. A key downside of this approach is the large number of proposals generated. This will lead to excessive processing times or call for a very simple classification method at the cost of recalling fewer hands. A typical optimization involves increasing the pixel offset as the window moves. An increased offset has as side effect that the probability of the window overlapping with a hand is reduced, thus the classification method may never consider the entire hand.

A single object of interest usually contains a couple forms of perceptual commonality. This could be single color, a color gradient or a particular texture. Felzenszwalb and Huttenlocher [13] leverage these forms of commonality to generate proposals from an input image by segmenting the input image into perceptual uniform regions. Starting with one pixel per region, their algorithm recursively merges neighboring pixels with similar color intensity until all pixels are contained in a region. To determine similarity, two factors are weighted; 1) the dissimilarity of elements along the boundary of the two regions, 2) the minima of the differences measured within each region. The algorithm has one key parameter, the threshold k which is used to determine whether to merge similar regions. The choice for k implicitly determines the size of resulting regions. Uijlings et al. [53] extend on this work by providing more diversification strategies to generate more proposal regions. Their algorithm is called Selective Search and uses various region similarity heuristics and color spaces to further merge the output of Felzenszwalb and Huttenlocher [13] their algorithm. A salient difference between the two algorithms is that Selective Search outputs overlapping regions and thus produces vastly more regions, but typically still fewer than an exhaustive sliding window approach. The following diversification heuristics are available for Selective Search to determine whether regions should merge:

- *Color* A similarity measure using histogram intersection, with 25 bins per color channel inside the considered color space.

- *Texture* A similarity measure through directional derivatives of a pixel in 8 cardinal directions. Each direction is quantized into a 10 bin histogram and uses histogram intersection for similarity.

- *Size* Merge similar sized regions. The size of a region is determined by the number of pixels contained. Uijlings et al. [53] suggest that this heuristic avoids one region from “gobbling up” all other regions when iteratively merging them.

- *Fill* Merge regions which fit well into each other. The goal is to remove regions containing a gap or merge well fitting regions into each other.

Multiple metrics can be combined to form a new metric. Selective Search is repeated per metric and per color space. The combined results from every Selective Search iteration are filtered for duplicates and shuffled randomly.

The pipeline proposed in this thesis will use Selective Search with various metrics and trend. The algorithm by Felzenszwalb and Huttenlocher [13] is not evaluated because many regions will also be contained in the results of Selective Search. A sliding window is not used because the number of proposals generated for an exhaustive search is impractical. Moreover, it seems justified to assume that a hand is found as a single region given its color, texture and because it likely stands out from its surroundings because of the painter’s composition.

4.2 Feature Extraction

The classification system needs a feature vector which encodes the hand proposal. A feature vector is essentially an n-dimensional vector that describes the characteristics of the hand proposal in a specific way. Multiple ways of encoding a hand can be concatenated into a single large feature vector. When doing so, it is important to normalize the magnitude of each dimension. This is to avoid the classification system becoming biased towards favoring similarity in inherently high dimensions, for example, a color dimension might be encoded using a byte $\in [0, 255]$ whereas a probability histogram typically uses a float $\in [0, 1]$ to store its value. The following two subsections discuss the descriptors used to generate feature vectors.

4.2.1 Colors

Colors are an often used cue to detect hands, e.g., [64, 42, 20]. Colors can be encoded in various ways, including a histogram and as raw pixel values. A histogram places emphasis on occurrences of colors, whereas raw values factor in the spatial arrangement of colors within the hand proposal. The latter gives the classifier an opportunity to deal with the hand proposal’s rectangular shape, e.g., a pixel near the middle of the proposal is more likely to actually belong to a hand than one near the border.

Colors themselves can be represented using a variety a color spaces Phung et al. [42]. The color spaces considered for the feature vector are RGB, HSV and CIELab. These are selected due to being the common denominator among the work of other researchers. Each color space maintains chrominance and luminance information, which Phung et al. [42] indicate as being necessary for optimal performance. HSV is often praised for its reasonable invariance to light conditions in its hue channel due to a decomposition into one chrominance and two luminance channels. CIELab has two chrominance channels and one luminance channel. RGB does not make an explicit distinction.

The number of dimensions of the resulting feature vector must remain constant. To encode raw pixel values, the hand proposals are resized to a fixed size. The size directly affects the processing speed, e.g., a 32 by 32 pixel proposal already requires $3,072$ dimensions in the feature vector. A reduction in size also has the effect for variance reduction due to the usage of a bilinear resizing algorithm. The pipeline will evaluated using multiple sizes to determine the best performing one.

For a histogram the number of bins used is paramount. Imagine two very similar colors, if there are as many bins as there are unique colors these will never be considered equal. This issue is solved when there are sufficiently fewer bins than unique colors. This poses a trade-off, when the number of bins goes down so will the distinctiveness between dissimilar colors. The number of bins is also very dependent on the dataset, color space and similarity metric; Phung et al. [42] use 256 bins per channel successfully on a large dataset to detect skin, but indicate that 32 bins performed better when only a small subset was used, they also find that as the bin count goes up, the results become color space invariant for their Bayesian classifier. Jones and Rehg [28] found 32 bins to perform best, with 16 bins closely followed. Stergiopoulou et al. [49] successfully use 16 bins to detect hands through skin color in photos. The pipeline's performance will be evaluated with multiple bin sizes using at most 32 bins because the dataset is relatively small to justify a higher count.

4.2.2 Edges

Hands have a distinctive shape. This shape could be described with edges. Edges are defined as the boundary between two adjacent pixels where the color intensity significantly varies. To keep the number of edges the same as the number of pixels it is typical to compute the intensity difference between the neighbors of a pixel along an axis [10]. When this is done along both the horizontal and vertical axes, the result can be treated as a $2d$-vector with an angle and magnitude. When the magnitude is sufficiently low, the edge is simply not considered an edge.

There are several approaches towards encoding these edges in a feature vector. The simplest form is identical to encoding raw pixels, it literally stores the intensity differences along the given axis. This has as downside that the angle is lost. Instead the angle could be stored. Either way, this approach does not allow for a high degree of variance between similar hands before they are actually considered similar. Not only must the hand shape be similar, they must also be positioned identically within the hand proposal rectangle.

To allow some hand position and scale variance, Dense-SIFT [30] can be used. After resizing the hand proposal to a fixed size, Dense-SIFT uniformly subdivides the proposal into equal cells and computes the most dominant edge angle. This angle is then stored into the feature vector. The approach varies from typical SIFT implementations by regularly computing the angles rather than at keypoints. This makes Dense-SIFT more compatible with a wide range of feature vector classification systems, at a cost of storing more information. The Dense-SIFT approach is very similar to histogram of oriented gradients (HOG) approach. A HOG approach always stores a histogram of angles, whereas with SIFT that is up to the implementor. HOG also introduces the notion of generating higher order histograms by combining histograms from adjacent cells, this is also referred to as block-normalization [8]. This is an important step because it avoids cells with no discernable gradient from weighting equal to a neighboring cell that does have gradients. The pipeline proposed in this thesis will use HOG, and evaluate varying parameters for cell size and histogram bin count, the number of cells used to form a block will be 2 by 2 which Dalal and Triggs [8] found to perform well in their application of detecting people in photos.

Dense-SIFT and HOG are local descriptors due to their subdivision of a hand proposal. It is possible to generate a global descriptor with similar invariance characteristics. One such method is Context Descriptors [3]. These discriptors capture the angle and distance between every pair of edges within a single proposal. The benefit of this descriptor is its full invariance to rotation. The downside is the sheer magnitude of the feature vector and not every hand proposal will have the same number of edges, necessitating that either some edges are ignored or introduced such that the resulting feature vector always has the same dimensions.

4.3 SVM Classification

A support vector machine is often used to classify high dimensional data [6]. In the proposed pipeline this is a binary decision whether or not a proposal is a hand, the resulting decision is supplemented with a confidence score. This score can be thresholded to find a desired balance between recall and precision. All possible feature vectors combined form a feature space. The SVM is tasked with segmenting the feature space with a decision boundary such that any feature vector falls in the correct segment. There are several variants of SVM. The simplest form is a linear SVM, which bisects the feature space with a hyperplane. Feature vectors on one side of the hyperplane are classified as hand, and the opposite side as non-hand. The signed distance of the feature vector to the plane indicates the confidence score. It should be noted that the hyperplane is set to maximize the distance to each feature vector of a particular class. A linear SVM has as advantage that it is quick to train (i.e., find the decision boundary), but struggle when the feature vector dimensions are not linearly separable. A non-linear SVM, is one that uses a more complicated decision boundary. The boundary is determined by the kernel function. A typical non-linear SVM kernel is the Radial Basis Function (RBF) [24]. A RBF SVM is practically more akin to a "nearest neighbour (KNN)" classifier, but instead of weighting each dimension equally, this is done via fitting a gaussian curve. The former adds regularization, which allows the SVM to generalize over the training data.

The pipeline will use a RBF kernel, motivated by the recommendation of Hsu et al. [24] and their observation that a RBF performs at least as good as a linear SVM when parameters are tuned. Accordingly, the pipeline SVM parameters will be tuned.

4.3.1 Training

The SVM needs training data in order to determine the decision boundary. Training data is split into two categories, positive and negative examples. Positive examples are depictions of hands, and negative samples conversely so. Uijlings et al. [53] provide an example on how to use Selective Search in combination with a SVM. For positive examples they use annotated data. For negative examples they use proposals generated by Selective Search that have an overlap between 0.2 and 0.5. Overlap is computed by dividing intersection through union. Proposals that fall in that overlap range are considered "hard to classify correctly" i.e., are expected to lie close to the optimal SVM decision boundary. Because negatives vastly outnumber positives they remove any negative from the training data such that none have an overlap of 0.7 or greater with another negative. Furthermore they remove 50% at random to reduce the number of negatives.

In their example they iteratively use hard negative mining. Misclassified examples are identified and included as additional negatives in successive training iterations. Their system converges within two iterations, which they attribute to the quality of Selective Search proposals.

The pipeline in this thesis follows their example, with the addition of including any proposal with a ground truth overlap of greater than 0.7 as positive. The former appears to work well for the relatively small dataset used in this thesis.

4.3.2 Finding hands

To find hands, all Selective Search proposals are considered by the SVM. The resulting proposals are thinned using a greedy non-maximum suppression (NMS) algorithm which removes any proposals such that no proposals have an overlap of more than 0.5. NMS is necessitated because Selective Search can generate many proposals in the same area, with each proposal having varying sizes and position. The greedy algorithm will always keep the proposal with the highest SVM score. The overlap score of 0.5 appears to work well in practice, across all evaluated feature vectors. The remaining proposals and their associated SVM score are then combined with other cues as discussed in the remaining sections of this chapter.

4.4 DeeperCut CNN

DeeperCut is a human pose detector by Insafutdinov et al. [25]. The pose detector has state-of-the-art performance on detecting multiple persons in photos. At the core sits a convolutional neural network (CNN) which is used to the detect body parts: heads, arms, torso and legs. A separate algorithm combines these parts to form persons, following the probabilities generated by the CNN. The authors have conveniently pre-trained the CNN on their photo dataset and can immediately be applied to detect people in Renaissance era paintings. The application of a CNN trained in a different domain isn't uncommon. Westlake et al. [57] apply a similar strategy with a CNN trained on photos and applied on paintings, however, their CNN returns bounding volumes around persons, rather than fitting an articulated stick-figure. Westlake et al. [57] also do not provide a freely available implementation. Contrasting to Westlake et al. [57], the pipeline in this thesis will not apply fine-tuning of the CNN.

DeeperCut identifies arms by detecting a shoulder, elbow and a wrist. To use this information a score is computed by measuring the distance between a wrist and the center of a hand proposal. The score is normalized by the painting's size and a gaussian weighted decay is applied. The score is inverted such that 1 indicates near and 0 means far. Gaussian weighting is necessary because other cues also use gaussian weighting (e.g., the hand size based cue). Having each cue follow a gaussian distribution allows for a more meaningful combination of cues (e.g., weighted sum).

Because DeeperCut returns more than just hands, it will also be used in combination with spatial priors. This is discussed in the next section.

4.5 Spatial and Size Priors

A probabilistic model can be used to describe the likelihood of a hand being in given location. Statistical priors from Section 3.7 show that certain regions contain no hands whereas others frequently contain them. A prime example is the lack of hands near the edge of a painting. Using this information is a simple way to cull false-positives and will be used in the pipeline. To capture the distribution of hands continuously rather than discrete, a gaussian kernel is fitted to the training data. This is also done by Bambach et al. [2] to filter out unlikely hand locations using a smaller subset of training data.

Hand positions can also be modelled relative to the face. As shown in Figure 3.12, hands are typically below the face and within a certain distance. The success of this approach depends on whether information is known about the face's location. Without having to setup a separate pipeline for face detection, two pre-existing frameworks will be used. 1) DeeperCut as discussed in Section 4.4 and 2) the framework for object detection by Viola and Jones [56]. The readily available framework by Viola and Jones [56] comes with several pre-trained classifiers. For the purpose of the pipeline the high-precision moderate-recall configuration will be used, this configuration is also used by Taylor and Morris [51] in their system to generate skin-color distributions by detecting faces.

Selective Search does not offer an explicit way to control the size of proposals. The merge threshold parameter which throttles the urge for two regions to merge could be used to coarsely determine proposal size. A low threshold value would merge many regions, thus generating larger proposals eventually. This merging algorithm is based on the selective Search heuristics, not the actual size of a region; although one of the heuristics is sensitive to size, this is only there to create regions of similar size. To offer more control over the size of proposals, the joint probability of a hand having a certain width and height is will be modeled using annotation data. The width and height of a hand will be normalized using the painting's diagonal.

4.6 Combining Multiple Cues

The output of each pipeline component can be reduced to a confidence score, which can then be combined in several ways to form a better score. Tang et al. [50] does this by a weighted sum of HOG and color similarity measures, after which thresholding is used to detect persons. Roy et al. [43] use two sequentially placed CNNs to generate and filter hand proposals with success. Stenger [48] uses color and motion-based likelihood estimates to propose regions containing a hand, which are validated by a shape-based template matching system. Mittal et al. [36] use the confidence output of a skin color similarity descriptor and a part-based deformable model to create a feature vector which is classified by a linear SVM.

The pipeline in this thesis will also use a linear SVM as a way to final classify whether a proposal is a hand, based on confidence scores generated using various cues. This implicitly allows the linear SVM to automatically figure out how much each descriptor should contribute in a weighted sum. Moreover, the contribution of each descriptor can be inspected by studying the individual weights. Low absolute weights are indicative of a poor performing component.

The addition of a linear SVM adds a second training phase to the pipeline. The first phase, using the training dataset, generates the low-level SVM, heatmaps for priors and DeeperCut distance scores. The second phase, again using the training dataset, creates the full training feature vectors needed for the linear SVM. This training feature vector is populated using the heatmap data and SVM from the first phase, the DeeperCut cue remains unchanged from the first phase. The use the same training data for both phases adds a form of overfitting to the low-level SVM cue, i.e., the low-level SVM fits well to its training data, thus its confidence scores will likely either be $+1$ (hand) or -1 (no-hand) with very few nuances in between, it is not expected that the evaluation set will generate such polarising confidence scores. Overfitting effect will be less prominent with the heatmap based cues due to the addition of a gaussian convolution.

Chapter 5

Evaluation

This chapter discusses the performance of the pipeline. The first section details how the dataset is split into multiple subsets. Following that the theoretical maximum performance is computed based on Selective Search hand proposal generation parameters. The remaining chapters explain how each pipeline component is tuned for performance. The final section evaluates the tuned pipeline with a dataset that was not used for tuning.

5.1 Processing of Paintings

The dataset of 400 annotated paintings is evenly distributed across the categories Tenebrism, Academicism, Early Renaissance and High Renaissance. These paintings are split into two datasets with some overlap:

- *Evaluation* Consists of five subsets with 100 paintings each. Each subset represents a painting category, and additionally a special fifth category (*mixed*) with paintings randomly selected from across all categories.

- *Experimentation* This set is used to tune and tweak parameters of the pipeline's components. 100 paintings are randomly selected from the evaluation dataset. Extra care is taken to exclude paintings which are in the *mixed* category.

Each dataset subset is further divided into 50 paintings for training and 50 paintings for testing. Here training refers to (e.g.,) extracting spatial priors and SVM creation. This split between training and testing assures that the resulting pipeline performance is not subject to overfitting on a single dataset.

5.2 Validation Methods

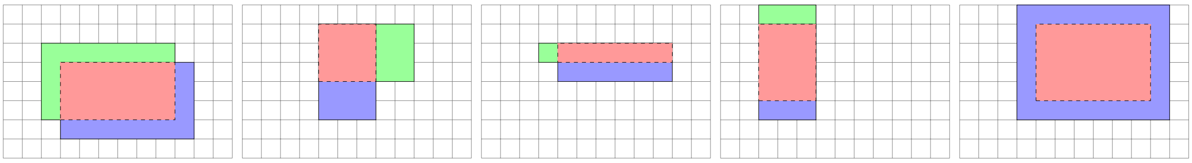
The Jaccard index is used to measure the similarity between a proposed hand and ground truth annotations. Proposed hands an annotations are axis aligned rectangles bounding a set of pixels. The Jaccard index is computed by dividing the overlap of the two rectangles by the area spanned by the rectangles. This approach is also known as *intersection over union* (IoU). The Jaccard index is useful because it factors in both the size difference of rectangles, as well as the offset between rectangles. It is bound on a $[0, 1]$ scale, with 0 indicating no overlap, and 1 indicating identicality. To get an indication of the effect of various overlaps, refer to Figure 5.1. It is observed that this statistic heavily penalizes errors, to the point where a 0.5 overlap can already be considered a good enough match, as is done by (e.g.,) [12].

The results of the pipeline will be further aggregated using the following statistics:

- *Precision* (P), the percentage of proposed hands that actually correspond to a ground truth hand. A hand is considered detected when it has a Jaccard index of at least 0.5 and the SVM score sign is positive, thus suggesting it is a hand.

- *Recall* (R), the percentage of actual detected hands versus ground truth hands. Describes how many hands are discovered, this statistic is not influenced by false-positives, but is influenced by hands not found (false-negatives).

- *Average Precision* (AP), the area under the curve generated by mapping precision against recall with a varying parameter such as a threshold. The resulting number aggregates a set of precision and recalls values into a single ordinal value which describes the shape of the curve. Higher values are better. When this is applied to multiple paintings, the average of all AP values will be used. This is sometimes referred to as mean-AP (mAP). The implementation used does not interpolate.



(a) 0.474 *(b) 0.429* *(c) 0.462* *(d) 0.667* *(e) 0.500*

Figure 5.1: Visualization of various Jaccard indices. The red dashed region indicates overlap. A Jaccard index of 1 means complete similarity, 0 means no overlap. For validation purposes 0.5 *or higher will be considered a good enough match.*

5.3 Proposal Generation

The authors of Selective Search propose three convenient presets using various color spaces, k values to determine initial region size, and their heuristics for color (C), texture (T), size (S) and fill (F), when multiple heuristics can be combined through multiplication to form a new one. Table 5.2 lists the parameters of each preset. Each preset offers a different balance between the quality of the produced proposals versus the number of proposals. Besides typical color spaces, the following abbreviations are used: (H) the Hue channel from HSV, (I) the gray scale representation, (rgI) the red and green colors from normalized RGB along with (I). In their work they considered several more color spaces, but those did not improve the performance on their benchmark dataset.

Preset	**Heuristics**	**Color spaces**	***k* values**
Single	CTSF	HSV	100
Fast	CTSF, TSF	HSV, CIELab	50, 100
Quality	CTSF, TSF, F, S	HSV, CIELab, rgI, H, I	50, 100, 150, 300

Figure 5.2: Selective Search presets as identified by Uijlings et al. [53].

To determine which strategy to use in the pipeline, an experiment is held to compare the performance of the presets. Per hand, the best matching Selective Search proposal is kept, subject to a 0.5 threshold on the Jaccard index. Table 5.3 lists the results. Each preset offers a trade-off between recall and the number of proposals generated. The Quality preset performs the best by recalling 89% of the hands with an average Jaccard index of 0.759, closely followed by the Fast preset with 77% recall and an 0.71 average Jaccard index. The performance drop by the Fast preset is offset by a 79% drop in proposals generated. Figure 5.4a shows the distribution of Jaccard indices for each preset. It shows the poor performance of the Single preset compared to Fast and Quality.

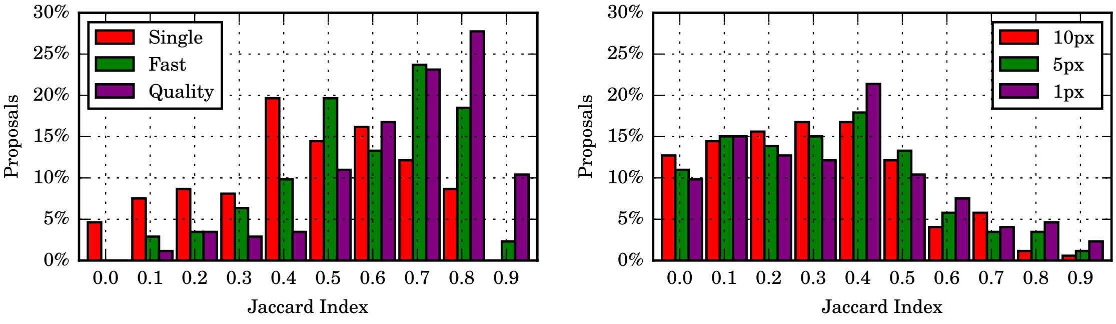
The negatives column in Table 5.3 indicates the average number of training samples per painting available when using the selection procedure as recommended by Uijlings et al. [53], i.e, negatives are proposals with an ground truth overlap on the interval $[0.2, 0.5]$. The latter does not account for filtering duplicates and randomly dropping 50%. The positives column indicates additional positive training examples available with a ground truth Jaccard index of at least 0.7. Given the balance between the number of proposals and their quality, the pipeline will be evaluated using the Fast preset. Fewer proposals will significantly improve processing speeds, and likely reduce the number of false-positives. It is accepted that this limits the pipeline hand recall upper bound to approximately 77%.

To relativise the performance of Selective Search, consider that a sliding window approach only has 29% recall with a window size equal to the average hand size and a stride of 1 pixel, as summarized in Table 5.3 with varying sliding window offsets. Table 5.4b shows the distribution of Jaccard indices. It appears that sliding window performance is invariant to the considered strides of 1, 5 and 10 pixels, suggesting that the poor performance is due to the fixed window size. The window size could be varied to improve performance, but this will linearly increase the number of proposals generated. At any rate, a sliding window is likely to underperform compared to Selective Search.

It is noted, that on the dataset used by Uijlings et al. [53] they achieve on average a 20% higher Jaccard index with near perfect recall. The latter may hint that Renaissance paintings are a more challenging dataset, especially because hands are relatively small with respect to the size of a painting. Uijlings et al. [53] also include categories with smaller objects such as bottles and birds, however, these photos are often composed in such a way that even small objects cover a large part of the image. Moreover, due to the size based merging criteria, Selective Search may favor creating proposals that are larger than a typical hand.

Preset	**Avg. Jaccard**	**Recall**	**Proposals**	**Positives**	**Negatives**
Single	0.674	51%	1,135	1	19
Fast	0.710	77%	5,521	7	128
Quality	0.759	89%	26,513	34	670
Sliding 10px	0.646	24%	2,376	< 1	2
Sliding 5px	0.660	27%	9,329	< 1	2
Sliding 1px	0.677	29%	229,714	< 1	2

Figure 5.3: Selective Search statistics averaged over the experimentation dataset. The recall column indicates the percentage of hands that have a proposal with a ground truth Jaccard index of at least 0.5.



(a) Selective Search

(b) Sliding Window

Figure 5.4: Jaccard indices produced by Selective Search and a sliding window. Jaccard indices are rounded to the nearest tenth.

5.4 SVM classification

There is a plethora of parameters that can be tuned to optimize SVM performance. Parameters can be broken into two categories, the individual descriptor parameters as discussed in Chapter 4 and parameters pertaining to the SVM and its kernel.

The SVM parameters are left at the default values as implemented by the Python language bindings for LIBSVM [40]. Notably this includes a class weighting such that both positives and negatives are represented equally during training. This is important because there are significantly more negatives than positives, thus will avoid a SVM bias towards classifying everything as “no hand”. The SVM will use a radial kernel (RBF). This kernel has two parameters. 1) the slack (c) value which determines the severity of misclassified data, i.e., those that lie on the wrong side of the decision boundary. A low c value allows the SVM to find a less-complicated boundary and will be easier to fit. A high value attempts to perfectly separate both classes, which may lead to overfitting. 2) the gamma (γ) parameter which controls the level of influence of a single feature vector. High γ values approximate a KNN classifier and are expected to overfit. Low γ values smoothen the border complexity up to the point where it is either linear or quadratic (i.e., polynomial kernel). Figure A.8 visually demonstrates the effect of parameters for a 2-dimensional toy-example. The choice for either parameter is arbitrary [24] and must be discovered through a search. The solution space of the search is continuous, but contains many local maxima, which necessitates the need for a grid search. Hsu et al. [24] suggest coarsely searching for initial parameters using $c = (2^{-5}, 2^{-3}, \ldots, 2^{15}), \gamma = (2^{-15}, 2^{-13}, \ldots 2^{3})$ and then finely searching the solution space around promising parameter pairs. A parameter set is considered better when it has a higher AP score.

Following from the above, well over 1,000 experiments would need to be run per descriptor (e.g., also factoring color space and bin size) before the fine search can be begin. This is very impractical. The following approach will be used instead: using $c = 1$ and $\gamma = 1/dimensions(feature\ vector)$ each individual descriptor will be evaluated with applicable parameters (e.g., color space). The most promising descriptor parameters are retained and repeated with varying SVM radial kernel parameters. The initial values for c and γ lie inside the suggested search range and on neither side of the extrema. It is hoped that this gives an indication of which descriptors and parameters perform well, i.e., a decision boundary easily found, prior to engaging an exhaustive search. The following sections discuss the performance of the individual descriptors and the choice of descriptor parameters.

5.4.1 Raw Pixels

The raw pixel descriptor encodes colors directly into the feature vector. There's a choice of color space and the fixed size to which proposals are resized. For width and height the sizes $(4, 8, 12, 16, 24, 32)$ are considered per color space. Table 5.1 lists the results. The best performing configuration in terms of AP (0.110) is the 4x4 RGB descriptor. It has the lowest precision and one of the best recall values, suggesting that this configuration benefits from classifying most proposals as a hand (albeit incorrectly) but with maintaining correct ordering SVM score. The fit on the training set achieves a much higher AP of 0.739, suggesting that the SVM fails to generalize on the provided hands. As the proposal size shrinks the raw color descriptor will approach a mean color, it could be that a small subset of hands all share a similar mean, on which the SVM discriminates. The performance of fitting on the training set increases as the number of pixels increases, a symptom of overfitting on training data.

Tuning the best performing 4x4 pixels feature vector improves the AP from 0.110 to 0.113, which slightly reduces the numer of false-positives. This is also seen in the precision recall curve in Figure 5.6a. The used parameters are $c = 2^{-1}, \gamma = 2^{-5}$. Figure 5.5a summarizes the performance of varying parameters, there is a trend that higher λ values perform much worse. The first column in Figure 5.7 shows the true positives and false positives, the SVM appears to be keen on accepting proposals containing the red and pink hues.

4x4 pixels is quite a reduction compared to the average hand size of 45x43 pixels. To confirm whether this is a mere fluke caused by the initial SVM kernel parameters, an additional tuning session was performed using 16x16 and 32x32 pixels. Both configurations still perform worse, with 0.111 AP ($c = 2^{-1}, \gamma = 2^{-13}$) and 0.09 AP ($c = 2^{7}, \gamma = 2^{-15}$), respectively.

	Using Testing Data			*Using Training Data*		
Parameters	**AP**	**P**	**R**	**AP**	**P**	**R**
RGB						
4x4	0.110	0.95%	47.25%	0.739	36.30%	85.50%
4x4 *(tuned)*	0.113	0.99%	44.74%	0.772	38.99%	85.71%
8x8	0.075	0.94%	37.65%	0.893	54.61%	93.20%
12x12	0.069	0.91%	32.96%	0.925	61.20%	92.70%
16x16	0.076	1.11%	36.33%	0.936	62.32%	95.02%
24x24	0.066	1.09%	35.25%	0.926	61.93%	94.49%
32x32	0.065	1.10%	34.42%	0.927	62.78%	94.49%
CIELab						
4x4	0.067	1.61%	44.48%	0.831	43.16%	89.50%
8x8	0.064	1.54%	35.95%	0.957	66.64%	94.88%
12x12	0.063	1.77%	36.92%	0.958	69.76%	95.17%
16x16	0.068	1.87%	34.95%	0.965	70.29%	94.88%
24x24	0.061	1.93%	37.62%	0.967	73.44%	95.17%
32x32	0.057	1.96%	36.72%	0.965	73.89%	94.88%
HSV						
4x4	0.038	0.99%	50.35%	0.766	41.83%	91.97%
8x8	0.056	1.10%	38.10%	0.914	57.96%	94.48%
12x12	0.068	1.04%	34.10%	0.952	65.89%	94.31%
16x16	0.075	1.03%	32.06%	0.952	67.07%	96.20%
24x24	0.065	1.05%	34.23%	0.950	68.17%	94.62%
32x32	0.065	1.55%	31.08%	0.960	70.72%	94.12%

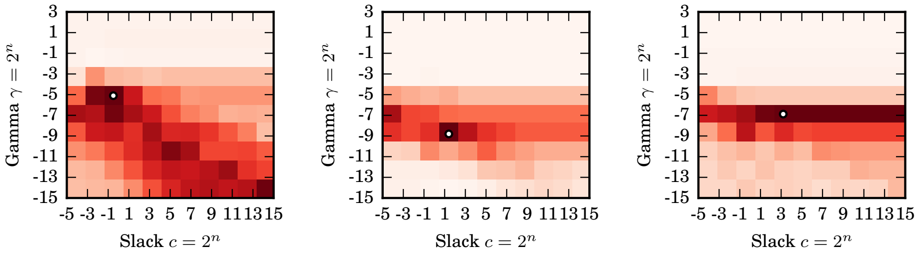
Table 5.1: SVM results using raw pixel descriptors.

5.4.2 Color Histograms

Color histograms offer a choice of bin quantity and color space. A low bin count reduces the discriminative factor, whereas a high bin count reduces the ability of a SVM to generalize. The bin quantities $2, 4, 8, 16, 32$ are evaluated. Bin quantities larger than 32 are impractical due to their sheer size, e.g., 32 bins per dimension already yields a feature vector with $32, 768$ dimensions.

Table 5.2 lists the results. Color histograms are outperformed by the raw color descriptors for most configurations. The best performing parameters are RGB with 16^3 bins, with 0.029 AP and a meager 0.29% precision. The tuned SVM kernel parameters improve the AP to 0.04 with improved precision, but reduced recall, using $c = 2^1, \gamma = 2^{-9}$. A general trend, Figure 5.5b, shows that γ reduces performance when on either end of its extrema. Performance remains reasonably consistent for any value of c.

The second column in Figure 5.7 shows that the descriptor accepts fewer Selective Search hand proposals than the tuned raw pixel descriptor, but in doing so reduces the number of true positives, leading to an overall reduced AP score. Comparing the average precision recall curve, in Figure 5.6 shows that among the SVM score ranking, hands are typically discovered further down in the ranking.



(a) Raw Pixels *(b) Color Histogram* *(c) HOG*

Figure 5.5: SVM classification performance for various kernel parameters. The color intensity runs from white to red, with red indicating the highest value. The range of AP values is [0.004, 0.113]. The white marker indicates the selected best performing configuration.

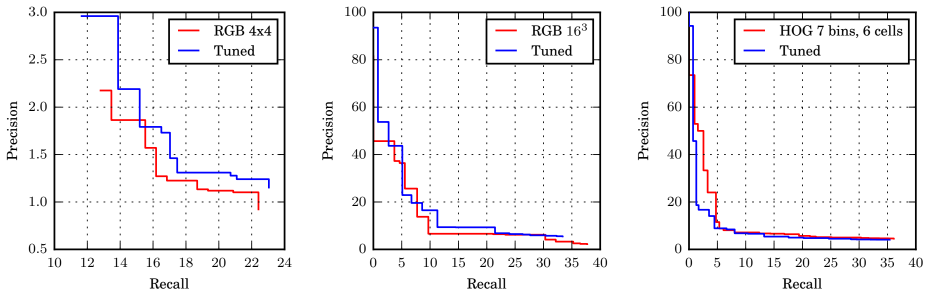
5.4.3 Histogram of Oriented Gradients

The HOG descriptor used [26] offers several configurable parameters. Notably the number of cells and the number of histogram bins per cell. The size of a cell will be deduced from the number of cells and the size of a proposal. A grid search is performed to find the best parameters. With the number of cells ranging from 2 to 8, and histogram bin number ranging from 4 to 20, only even cell numbers are evaluated. The cell histograms will concatenate their east, south and south-east neighboring cells to form a single block. Other parameters such as gamma (color intensity) correction, bin magnitude clipping and gaussian smoothing are left at default.

Table 5.3 lists the results of the grid search for parameters, with some low performing results omitted for brevity. The best performing configuration with 7 bins and and 6x6 cells, achieves 0.085 AP. By tuning the SVM kernel parameters the AP increases marginally to 0.094, but the recall drops from 19.64% to 4%, with a gain in precision from 2.69% to 5.33%. The best kernel parameters are $c = 2^3, \gamma = 2^{-7}$. The general performance trend is the same as with color histograms, showing reduced performance for high and low γ values.

5.4.4 Combining Descriptors

The feature vectors from raw pixels, color histograms and HOG can be concatenated into a single feature vector. The resulting feature vector is normalized such that the mean is nought and variance is unity. This removes any implicit bias towards favoring larger numbers (e.g., colors) over smaller numbers (e.g., a probability measure). Using default SVM kernel parameters the AP becomes 0.092 with 3.18% precision and 38.60%



(a) Raw Pixels

(b) Color Histogram

(c) HOG

Figure 5.6: Average precision recall curves for each descriptor before and after tuning. The curves are generated per painting using 41 uniformly spaced SVM score thresholds on the interval [-1, +1], and then averaged across all involved paintings per threshold. The resulting average precision scores are visually linearly interpolated as described by Manning et al. [35].

recall. Tuning the SVM parameters, the AP becomes 0.103 with a precision increase from 2.37% to 3.18% and a recall increase from 38.15% to 38.6%. This tuning process includes an additional fine grid search using $c = (2^2, 2^{2.25}, \ldots, 2^4), \gamma = (2^{-10}, 2^{-10.25}, \ldots 2^{-12})$, the best performing parameters are $c = 2^{1.5}, \gamma = 2^{-11.5}$ (Figure 5.9).

The performance in terms of precision and recall sits between the other individual descriptors. Figure 5.7 shows the implications, the combined descriptor accepts fewer proposals than the raw descriptor, but significantly more than the HOG and color histogram descriptors. This is also reflected in terms of AP where the combined descriptor performs worse than raw pixels, but better than the others.

Table 5.8 indicates the overlap between the hands found of individual descriptors and the combined descriptor. A hand is considered found when the SVM indicated a positive score and the ground truth Jaccard index is at least 0.5. The biggest contributor towards hands found by the combined descriptor are raw pixels, with some 67% of the hands also present in the individual descriptor. All of the hands found by the HOG descriptor are still present in the combined descriptor, although this is not as impressive considering the low recall value of HOG, which is reflected by the combined descriptor containing merely 7% of the hands also found the HOG on its own. 21% of the results in the combined descriptor are not found by any other individual descriptor, suggesting that combining descriptor effectively provides new information for the SVM to generalize on.

	Using Testing Data			*Using Training Data*		
Parameters	**AP**	**P**	**R**	**AP**	**P**	**R**
RGB						
2^3	0.019	0.74%	11.79%	0.492	30.34%	70.09%
4^3	0.023	0.26%	10.47%	0.343	19.81%	71.83%
8^3	0.013	0.29%	9.67%	0.376	20.26%	66.50%
16^3	0.029	0.28%	11.00%	0.287	15.46%	63.88%
16^3 *(tuned)*	0.040	0.39%	4.29%	0.306	22.65%	70.99%
32^3	0.012	0.66%	10.16%	0.582	32.56%	72.57%
CIELab						
2^3	0.012	0.05%	4.00%	0.271	11.82%	66.17%
4^3	0.016	0.14%	5.13%	0.279	17.11%	76.54%
8^3	0.008	0.13%	7.47%	0.300	16.17%	66.58%
16^3	0.017	0.25%	8.80%	0.315	18.41%	64.41%
32^3	0.018	0.26%	9.72%	0.409	19.82%	84.41%
HSV						
2^3	0.011	0.19%	7.50%	0.244	10.55%	42.51%
4^3	0.015	0.33%	11.57%	0.388	23.26%	66.49%
8^3	0.008	0.11%	4.00%	0.477	26.61%	67.23%
16^3	0.015	0.36%	11.31%	0.642	31.88%	73.51%
32^3	0.012	0.30%	10.81%	0.842	61.88%	83.80%

Table 5.2: SVM results using color histogram descriptors.

Descriptor	**Combined**	**Color Hist.**	**Raw Pixels**	**HOG**	**Unique**
Combined	-	19%	67%	7%	21%
Color Hist.	80%	-	50%	10%	0%
Raw Pixels	49%	8%	-	5%	47%
HOG	100%	33%	100%	-	0%

Figure 5.8: Indication of distinct hands found per tuned descriptor. Table is to be read as "How many hands in set A (row) are also contained in set B (column)?".

5.4.5 Hard Negative Mining

The number of negative examples is reduced to benefit training speed of the SVM. 50% of the negatives are randomly dropped and similar negatives are removed if their Jaccard index is more than 0.7. Due to the reliance on randomness, it is not guaranteed that this subset of negatives is optimal for training. Hard negative mining [53] is performed to tune the training dataset. The hard negatives are obtained by evaluating the unfiltered training dataset, and adding them to the filtered training set.

This retraining process offers a choice of quantity of hard negatives and the weight of hard negatives. The weight determines how much effort the SVM should place on accommodating a particular feature vector, in addition to the slack c value. The number of hard negatives sampled per painting is

	Using Testing Data			*Using Training Data*		
Parameters	**AP**	**P**	**R**	**AP**	**P**	**R**
6 bins						
2 cells	0.049	0.19%	13.67%	0.748	39.36%	86.57%
4 cells	0.056	1.34%	32.11%	0.983	83.48%	96.50%
6 cells	0.058	1.60%	19.02%	0.993	97.18%	99.29%
8 cells	0.066	2.66%	16.16%	0.999	98.93%	100.00%
7 bins						
2 cells	0.047	0.22%	15.67%	0.719	40.00%	83.29%
4 cells	0.060	1.53%	36.43%	0.986	85.85%	97.33%
6 cells	0.085	2.69%	19.64%	0.999	97.60%	100.00%
6 cells *(tuned)*	0.094	5.33%	4.00%	1.000	100.00%	100.00%
8 cells	0.068	3.54%	14.80%	0.999	100.00%	100.00%
8 bins						
2 cells	0.041	0.23%	17.33%	0.792	43.57%	83.95%
4 cells	0.064	1.70%	31.11%	0.982	90.30%	97.67%
6 cells	0.069	2.31%	15.12%	0.998	97.17%	100.00%
8 cells	0.073	3.46%	12.05%	1.000	100.00%	100.00%

Table 5.3: SVM results using Histograms of Oriented Gradients. Only well performing configurations are included. In total 68 parameter pairs where considered.

evaluated for $(1, 2, 3, 5, 10, 50, 100)$ as well as an *all* option, which includes all false positives. This is repeated using varying weights $(1, 5, 10, 100)$, where weight 1 is the default of all feature vectors.

The best performing configuration achieved an AP of 0.098 which is a marginal decline over 0.103. The precision dropped from 3.18% to 2.07%, recall improved from 38.6% to 39.5%. This setup used 5 samples, and is invariant to sample weight. Figure 5.10 summarizes the performance across the evaluated parameters. The drop in performance is unexpected, other researchers noted performance improvement on their datasets, e.g., [53, 8, 14]. One potential explanation is loss of generalization due to overfitting. Figure 5.11 shows a strong increase in the number proposals classified as hand by the SVM when more hard negatives are added. The figure includes the top 500 proposals most likely to be a hand from each painting.

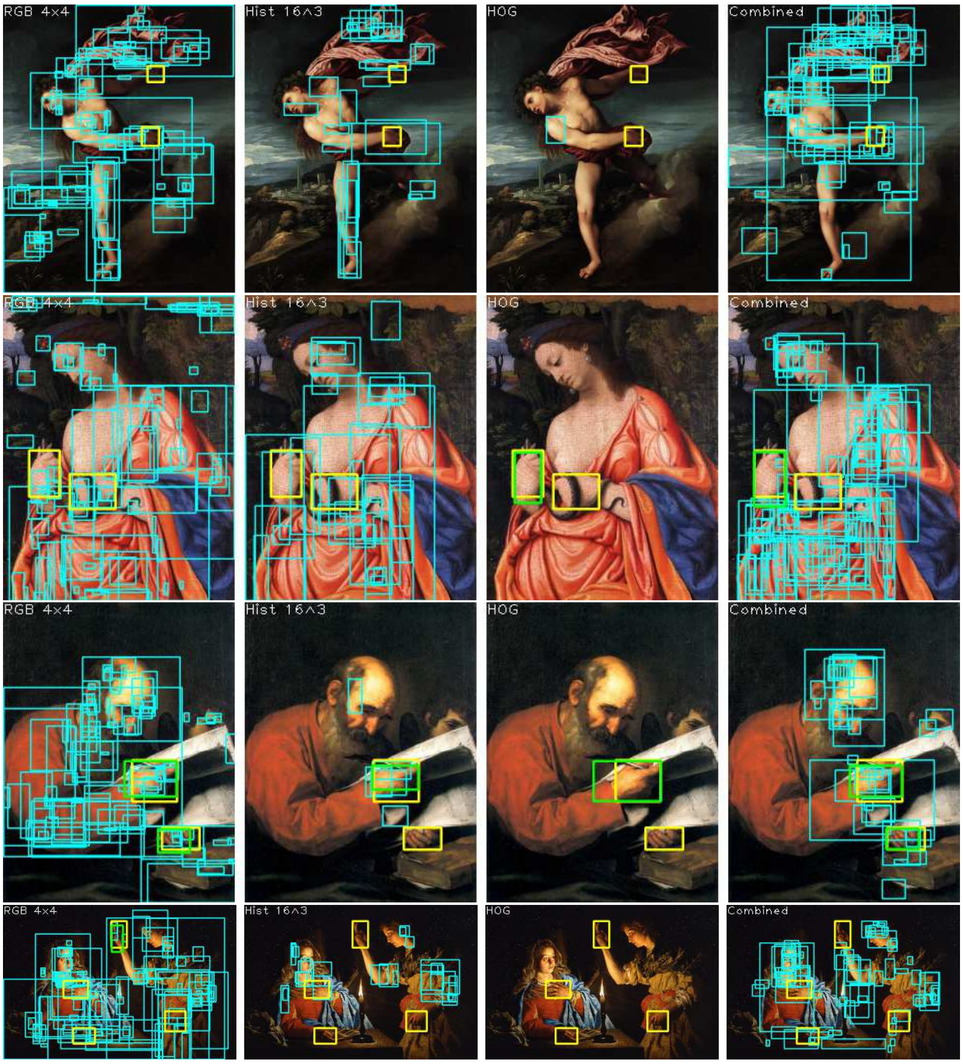
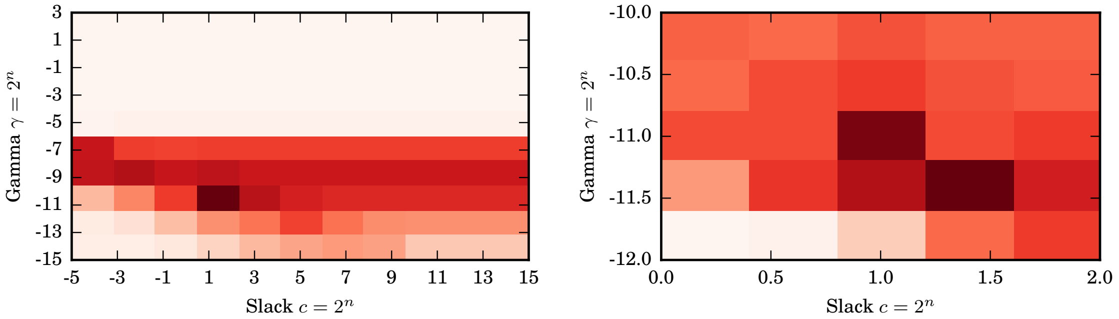


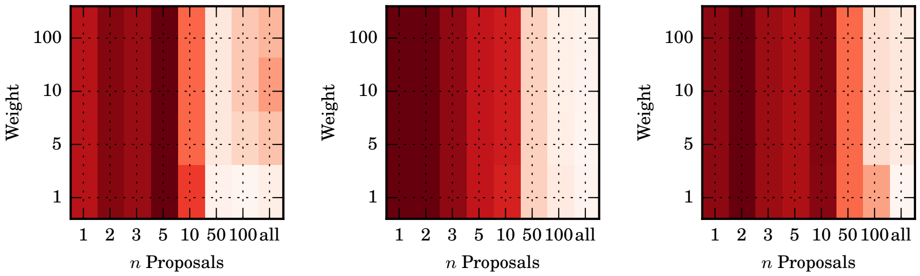
Figure 5.7: True positives (green) and false positives (cyan) generated by the tuned SVMs. Ground truth is shown in yellow. From left to right: RGB 4x4 pixels, RGB histogram 16^3, *Histogram of Oriented Gradients, and, all the descriptors combined.*



(a) Coarse search

(b) Fine search

Figure 5.9: SVM classification performance of combined descriptors. The color intensity runs from white to red, with red indicating the highest value. The range of AP values is [0.004, 0.103].



(a) AP

(b) Precision

(c) Recall

Figure 5.10: SVM classification performance by including additional hard negatives. The AP value range lies in [0.049, 0.098], recall [10, 39.15], precision: [0.47, 2.37].

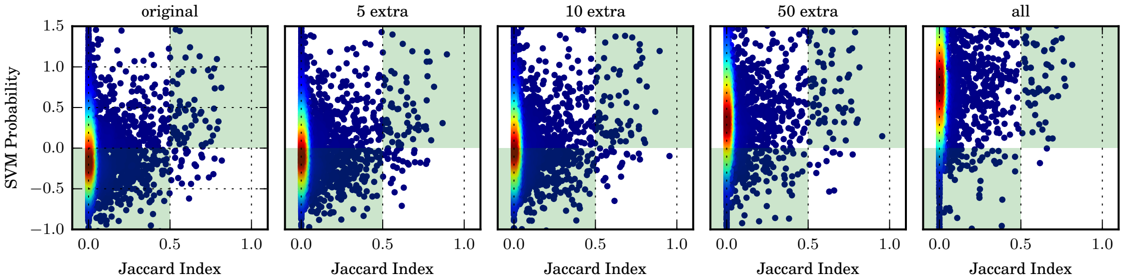


Figure 5.11: Correlation between SVM confidence score and Jaccard Index. From left to right with an increasing number of hard negatives added, negatives are default weighed. NMS is used to thin the results. A probability density kernel is fitted (KDE) and reflected through the colors.

5.5 DeeperCut Performance

DeeperCut processing of a single painting takes about 40 minutes on an Intel Core i7 2630QM with 8GB RAM, with each painting downscaled to have no dimension exceed 512 pixels. The downscaling is required by hardware constraints. The computer's GPU could not be used due to insufficient VRAM available. The DeeperCut parameters are all left at default. Figure 5.14 shows some typical DeeperCut results when applied to paintings.

To measure the best-case potential of DeeperCut, each detected wrist is mapped to its closest ground truth hand, where the center of the hand is used as anchor point. Subsequently a rectangle is centered on the wrist location using the width and height of the nearest ground truth hand. The Jaccard index is computed between this new rectangle and the ground truth.

This process shows that 7% of $1,313$ DeeperCut's wrist proposals are within 0.5 overlap of a hand. This number vastly improves to 23% when exploiting the assumption that a hand's center lies along the axis running through the elbow and wrist. Using this combined information the DeeperCut hand proposal was translated along said axis with varying offsets $\frac{1}{8}, \frac{2}{8}, \cdots \frac{7}{8}$ proportional to the distance between the elbow and wrist. Offset $\frac{6}{8}$ performed the best. Table 5.13 details the results per painting category. Tenebrism performs the worst with 18% precision and Early Renaissance the best with 25% precision. The average recall across all styles is 20%. Performance on Academicism is not the best by either recall or precision, which is surprising given the often realistic nature of those paintings.

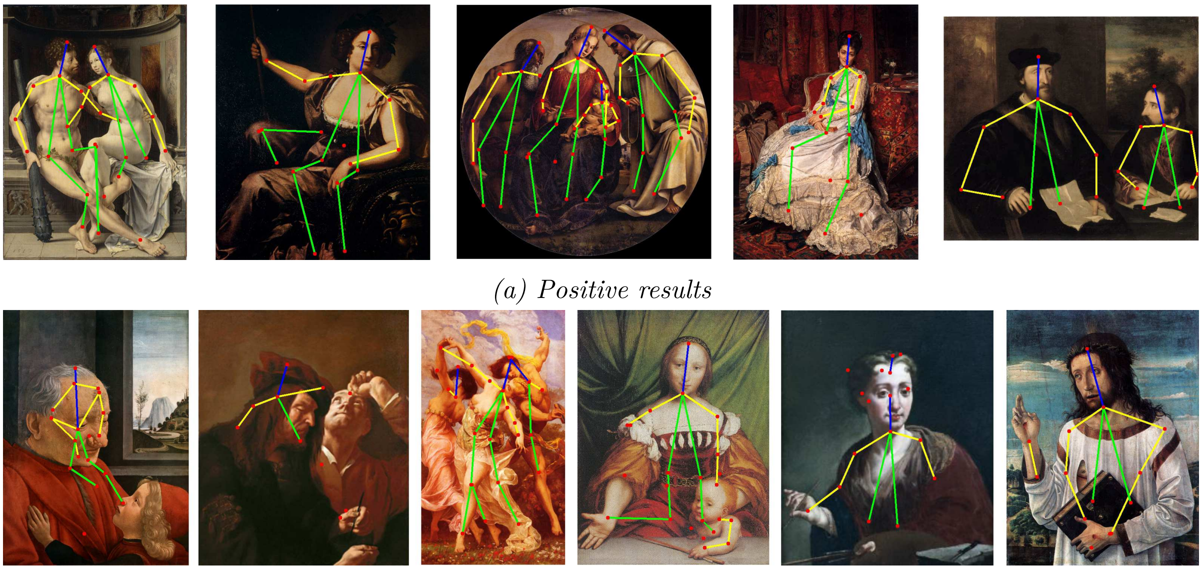
The precision remains reasonably consistent with respect to the number of persons in a painting. The highest precision, 25% is achieved with two persons, the lowest with a single person at 20%. The latter may be explained by the increased number of torso only portraits, which at times lead DeeperCut to find limbs at arbitrary locations such as inside a beard or background. Table 5.12 shows a breakdown. The number of proposals gets reduced as the number of persons goes up, likewise so does the recall. Presumably the composition in a painting changes as the number of subjects increases, which may influence DeeperCut.

	Hands				Heads			
Persons	**GT**	**Proposals**	**P**	**R**	**GT**	**Proposals**	**P**	**R**
1	329	440	20%	26%	187	215	41%	48%
2	344	356	25%	24%	206	189	33%	31%
3	225	182	24%	19%	143	104	46%	34%
4	229	183	21%	16%	144	99	34%	24%
5	221	152	24%	15%	144	92	41%	26%

Figure 5.12: The number of hands and heads found by DeeperCut grouped by the number of persons in a painting.

	Hands				Heads			
Category	**GT**	**Proposals**	**P**	**R**	**GT**	**Proposals**	**P**	**R**
Tenebrism	341	349	18%	17%	210	193	31%	28%
Academicism	277	280	22%	21%	158	146	40%	37%
Early Renaissance	418	349	25%	20%	242	185	42%	32%
High Renaissance	349	335	24%	23%	213	175	43%	36%
All	1385	1313	23%	20%	823	699	39%	33%

Figure 5.13: DeeperCut performance on the Paintings Dataset for detecting hands and heads. GT refers to the number of ground truth entries. Proposals column indicates the number of heads or hands (wrists) proposed.



(a) Positive results

(b) Negative and partial results

Figure 5.14: Stick figures created by using limbs as detected by DeeperCut [25]. Arms in yellow, head blue and legs green. The detected body parts are colored red.

5.6 Spatial Priors

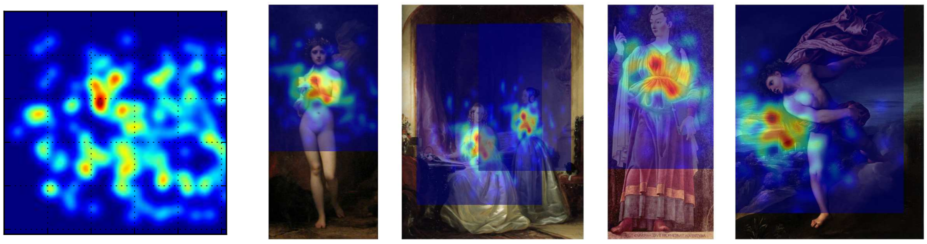
Spatial prior maps are generated using the training painting subsets. The global priors are encoded by normalizing the painting's size to unit space. Figure 5.15a shows an example generated from the experimentation subset, the resulting heatmap indicates a low probability around the edges of a painting, which is as expected. Because heatmaps are generated using just 50 paintings, they are convolved with a discrete gaussian kernel. The variance (sigma) of the gaussian is selected such that the area under the curve for 95% overlaps with the region spanning the average hand size, this approach worked well in practice. The convolution of the heatmap allows for some nuance, because it is not expected that the 50 training paintings fully represent all possible paintings in terms of hand positions.

Local priors are generated based on the hand's position with respect to the head. As with global priors a convolution kernel allows for nuance. Figure 5.15a shows how these priors look in practice by overlaying the heatmap onto a painting using annotated head positions and sizes. The local priors appear to be predominantly centered on the torso.

The admissibility of local priors hinges on the availability of head locations. Two frameworks are used to detect faces, DeeperCut and Viola-Jones cascade detector. DeeperCut successfully detects 42% of the ground truth heads with 49% of its head proposals correct. This follows from the same experiment as described in Section 5.5, except that the head's center is computed by averaging the position of the scalp and neck, if one of the body parts is missing, the other is used alone. For the purpose of local priors both the width and height of a head must be known such that the heatmap can be scaled accordingly. The height is measured using the scalp and neck position, if available. The width of a head can be estimated by using the average annotated head width/height ratio of 0.873. This approach, compared to using ground truth size information, reduces the precision to 39% and recall to 33%, but no longer relies on ground truth data. Table 5.13 lists these results aggregated per painting category. There's a similar trend compared to hand detection, Tenebrism performs the worst with 33% precision, and the others average about 10% higher. The number of proposals generated reduces as the number of persons per painting increasings (Table 5.12), this behaviour is consistent with wrist detection.

DeeperCut's usage in finding heads is much more successful than finding hands. This is by part because deducing a head position from a scalp and neck is much more accurate than basing it on elbow and wrist information. Also, DeeperCut uses spatial information to model the relative position of body parts, a head's position with respect to shoulders and hips has fewer degrees of freedom compared to an arm with respect to a shoulder. Moreover, DeeperCut is trained using photos of real world people, it is expected that head positions are more consistent across both domains than arms, e.g.,, in photos one expects arms to mostly fill a functional role, whereas in paintings they may be composed in such a way to express a gesture or convey a meaning.

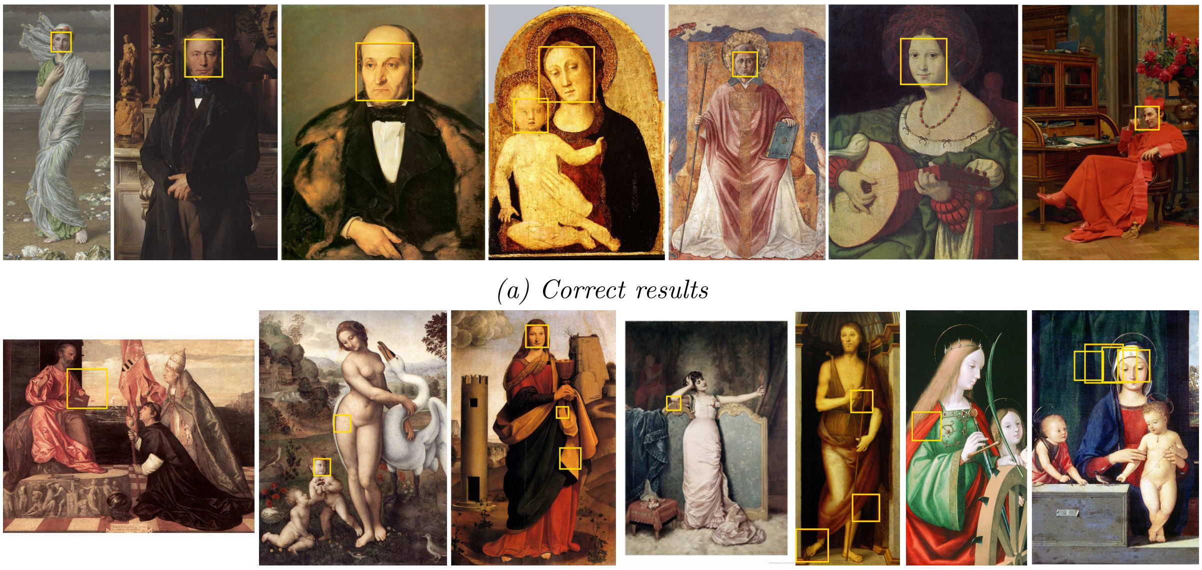
The other framework to detect hands is the cascade detector by Viola and Jones [56], which offers three available parameters. 1) The detector uses multiple image resolutions, the scale factor determines the downscaling ratio for each iteration. 2) The minimum number of neighbors required before a head proposal is produced. 3) The choice of decision tree. The high number of training examples required make it impractical to create a decision tree using faces from paintings (e.g., Viola and Jones [56] use 5000 faces). OpenCV ships with several community contributed pre-trained decision trees. Of these the *frontalface-alt2* was selected because in practice it showed the most promising precision to recall balance when compared to the ground truth subject to a minimum Jaccard index of 0.5. For the other two parameters a grid search took place. Table 5.17 lists the results of a grid search for the scale factor and minimum neighbor count. There is a strong correlation between a higher scale factor and reduced recall. The 83% precision and 17% recall configuration using 5 neighbors and scale factor 1.2 will be used due to its best precision to recall ratio. Figure 5.16 shows some of the typical results, performance is better for paintings where the person is frontally facing.



(a) Global Priors

(b) Local Priors

Figure 5.15: Hand position probability heatmap generated from the experimentation training dataset.



(a) Correct results

(b) False and partial results

Figure 5.16: Faces as proposed by the cascade classifier.

	2N		**3N**		**4N**		**5N**		**6N**		**7N**		**8N**	
Scale	P	R	P	R	P	R	P	R	P	R	P	R	P	R
1.1	0.46	0.23	0.51	0.21	0.61	0.17	0.52	0.20	0.65	0.16	0.31	0.23	0.56	0.18
1.2	0.73	0.20	0.78	0.17	0.82	0.13	0.82	0.17	0.82	0.13	0.56	0.22	0.85	0.16
1.3	0.76	0.17	0.88	0.14	0.88	0.13	0.88	0.14	0.93	0.12	0.58	0.19	0.88	0.14
1.4	0.65	0.14	0.75	0.14	0.86	0.06	0.76	0.12	0.83	0.05	0.60	0.17	0.73	0.07

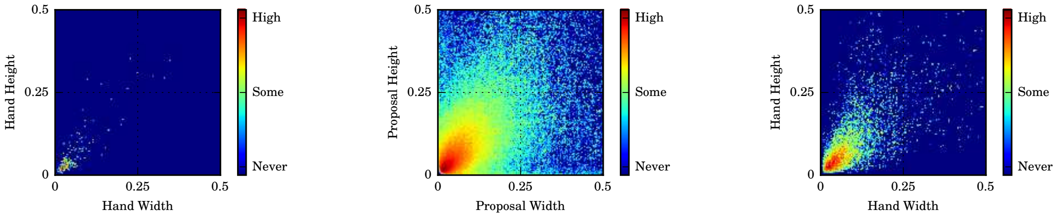
Figure 5.17: Precision and recall values of Viola-Jones cascade classifier [56] applied to paintings from the experimentation subset. Some low performing results are omitted for brevity.

5.7 Size Priors

Selective Search proposals will have the size of any region of interest in the painting. Figure 5.18b shows the spread of widths and heights when normalized by the painting's diagonal. This shows there are many proposals that have a width or height vastly exceeding the size of a typical hand. Hands are typically more squarish (Figure 5.18a), whereas the Selective Search proposal width/height ratios are much more spread.

Investigating the size of hands proposed by the SVM using the color histogram, raw pixels and HOG descriptor combination, shows that the SVM already does a great job at filtering out odd sized proposals where either width or height vastly exceeds the other. Figure 5.18c shows the resulting scatter plot. Even so, it is still apparent more results can be culled based on size alone, i.e., using prior information.

As with spatial priors, the probability map is convolved with a discrete gaussian kernel to allow for some nuance. The kernel variance (sigma) is half the value used with spatial priors, this appears to work well in practice due to the reduced variance in size with respect to variance in position.



(a) Ground truth *(b) Selective Search* *(c) SVM hands*

Figure 5.18: Width and height pairwise probability relative to a painting's diagonal. Results generated using paintings from the experimentation dataset.

5.8 Combining Multiple Cues

The output of each individual cue will be concatenated into a single feature vector. The cues are 1) DeeperCut hand distance (DC); 2) DeeperCut face detection with local priors (DF); 3) cascade face detector with local priors (VJ); 4) global priors describing hand absolute positons (G); 5) the hand width and height pairwise probability (S); and lastly, (6) the output from the low-level SVM (LL). The dimensions in the feature vector are normalized to zero-mean and unit variance. The latter removes any bias towards implicitly favoring optimisation of a particular cue because its value spans a greater range

The linear SVM (LSVM) needs positive and feature vector negative examples. These will be derived from the top 500 most likely hand proposals as generated by the low-level SVM training set. This subset of proposals already has a recall of about 90%. These proposals are assigned a positive label if their ground truth Jaccard index is at least 0.5, the remainder is considered negative i.e., non-hand. The cues are derived using the training subset.

Table 5.4 has results of each individual cue, as well as each cue paired up with the Low-Level SVM cue. An interesting result is that using just the LL cue, the performance already improves in terms of precision, going from 3.18% to 4.27% while maintaining 0.103 AP and a drop in recall from 38.60% to 27.30%. Out of all cues the LL+VJ and LL+S combinations performs the worst, when looking at the LSVM weights these cues each receive a weighting of |0.04|. To determine whether 0.04 is meaningful weight, consider that LL combined with a uniform distributed random number weights the random number with 0.03, suggesting that the VJ and S cue perform nearly on par with a random number. The best combination is LL+DC, which is also reflected in the individual performance, where DC (0.146 AP) outperforms LL (0.103 AP). The gaussian variance for DC was tuned to 2^{-5} after a grid search on $(2^{-10}, 2^{-9}, \cdots 2^{10})$. Although 2^{-5} may appear low, recall that distances are normalized to a $[0, 1]$ range.

When all cues are combined the AP is 0.202, with 7.13% precision and 37.82% recall. This is a great improvement from using just the tuned low-level SVM from Section 5.4.4. The lowest weight of 0.06 is attributed to VJ, however, when VJ is omitted both the precision and recall drop about 1% with AP remaining the same. This suggests that VJ does contribute one way or another, a random number would receive a weight of 0.01 when used in conjunction with the cues. The LL cue has by far the highest weight, interestingly, when the LL cue is removed, the AP drops to 0.173 which is worse than using just the DC cue, suggesting that the LL cue works well with the other cues. The worst impact occurs when the DC cue is removed, with AP dropping to 0.120. The drop in performance when either DC or LL is removed is not proportional to their LSVM weights, suggesting that solely looking at just the weights to determine a cue's importance is too naive. Furthermore, when the LL cue is omitted the LSVM bias number halves, suggesting that the high weight encodes a property intrinsic to the cue, e.g., it could be that variance among false-positives and true-positives is low, thus the extra weight is needed to make them separable after adding the other cues. When both DC and LL are omitted, the DF cue's weight doubles but overal performance drops to 0.053 AP. Table 5.5 summarizes the comparison of various cue combinations and their performance.

The LSVM kernel offers one tuning parameter, the slack parameter. As with the radial kernel, a grid parameter search was performed. The performance in terms of AP remained persistent for any slack value evaluated. To confirm whether this was due to some sort of rate-limiting, the SVM stopping criterion tolerance was buffed from $1e - 4$ to $1e - 6$ and the maximum number of iterations increased from 1000 to 100000, at no avail. The change in these parameters did increase training times significantly.

Figure 5.19 gives insight into the results of some cues when evaluated standalone. This is done by overlaying the low level SVM accepted Selective Search proposals using a color to indicate the cue specific likelihood. A few general observations can be made;

- DC DeeperCut performs well at locating proposals near a hand, but does not make a distinction based on the size of a proposal. On its own it performs better than the other descriptors.

- DF Positional priors based on DeeperCut detected faces performs better than the similar VJ cue. This is attributed to DeeperCut finding at least one head in every painting, and overal has a higher accuracy in doing so. It strongly prefers hand proposals which are on the person's torso. Presumably this bias is introduced due to the high number of Madonna holding her child Jesus, quickly adding 4 hands in a close proximity.

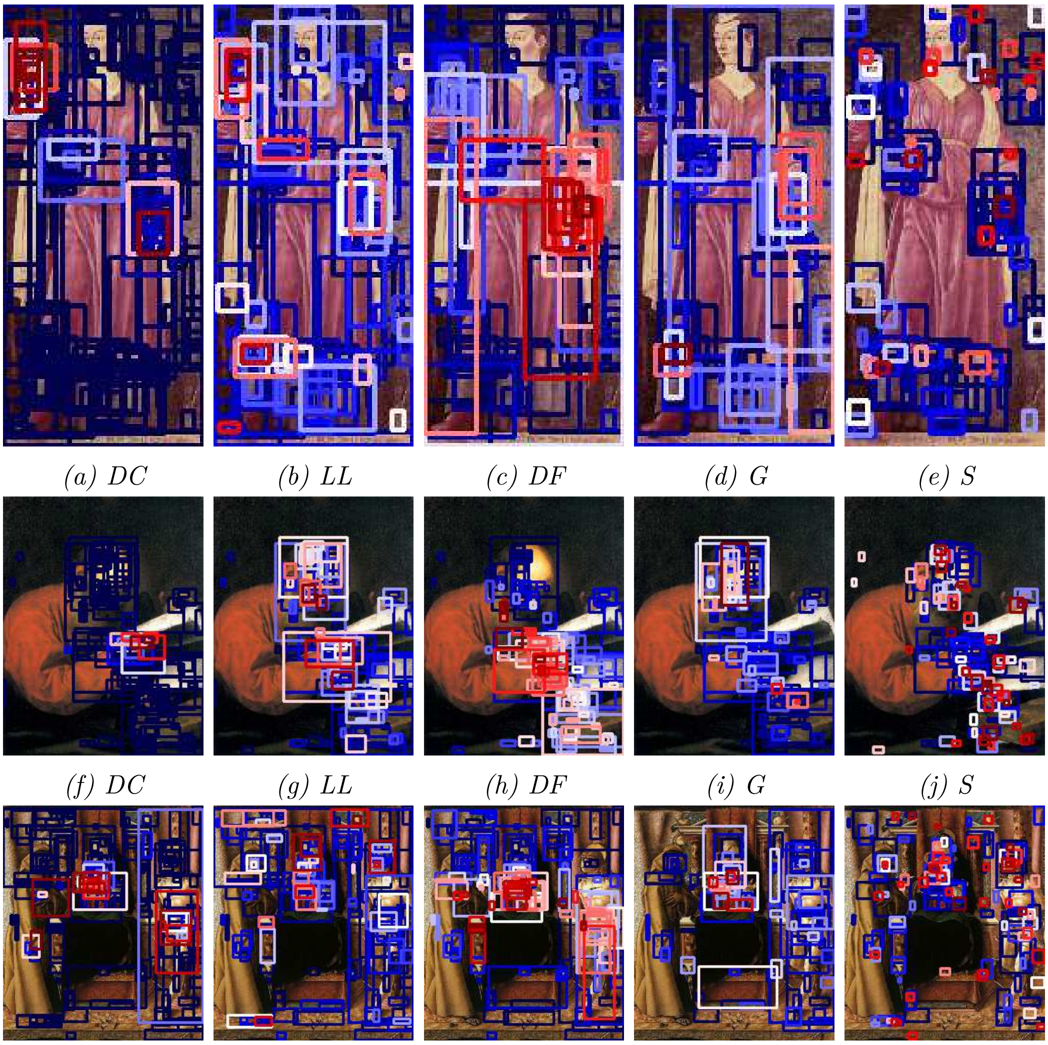
- S The hand size based prior performs a great job at selecting proposals which are small enough to be a hand, or have an appropriate aspect ratio. Table 5.5 indicates the LSVM weight is always negative, suggesting that this cue is used to remove hand-sized proposals from the results, i.e., it could be that another cue is generating too many hand-sized false-positives.
- VJ The cascade descriptor only finds a head in 30% of the paintings, compared to DeeperCut which proposes a head in all paintings. This leads to a reduced contribution of VJ towards the overall performance.
- G The global hand position prior adds little value, in general key descriptors such as LL and DC do not include hands near the edges of a painting. This cue also heavily discriminates against the sporadic hand near the painting boundary. The LSVM assigns a negative weight, suggesting this cue actually contributes to finding hands outside of common locations as suggested by prior data.
- LL The radial SVM finds objects of all sizes. This is because all its descriptors are size-normalized one way or another. It typically finds objects that are skin colored, including faces and patches of background.

	Standalone Cue			*Cue + LL*			LSVM Weights	
Cue	**AP**	**P**	**R**	**AP**	**P**	**R**	**LL**	**Cue**
-	-	-	-	0.103	4.27%	27.30%	0.44	-
DC	0.146	7.22%	32.12%	0.185	6.34%	37.82%	0.42	0.22
DF	0.045	2.63%	32.59%	0.117	5.22%	32.46%	0.41	0.21
S	0.019	1.79%	31.51%	0.107	4.47%	24.55%	0.44	-0.04
VJ	0.016	0.39%	8.77%	0.104	4.72%	29.02%	0.43	0.04
G	0.015	1.61%	40.50%	0.098	4.28%	25.24%	0.44	-0.06

Table 5.4: Performance of individual cues, as well as individual cues combined with the low level SVM (LL)

LSVM Weights						*Performance*		
DC	**DF**	**S**	**VJ**	**G**	**LL**	**AP**	**P**	**R**
0.21	0.11	-0.09	0.06	-0.08	0.39	0.202	7.13%	37.82%
-	0.21	-0.05	-0.01	-0.07	0.41	0.120	5.61%	32.46%
0.23	-	-0.09	0.10	-0.07	0.40	0.188	7.05%	37.08%
0.20	0.11	-	0.05	-0.08	0.40	0.192	6.40%	37.31%
0.20	0.14	-0.09	-	-0.08	0.39	0.202	6.61%	36.79%
0.21	0.11	-0.09	0.06	-	0.39	0.200	7.06%	37.82%
0.23	0.18	-0.13	0.08	-0.07	-	0.173	5.38%	36.55%
-	0.32	-0.07	-0.01	0.06	-	0.053	2.78%	28.98%
0.22	-	-	-	-	0.42	0.185	6.34%	37.82%

Table 5.5: Performance of multiple cues combined. A hyphen indicates that a cue is omitted from the feature vector when using the linear SVM.



(a) DC (b) LL (c) DF (d) G (e) S

(f) DC (g) LL (h) DF (i) G (j) S

(k) DC (l) LL (m) DF (n) G (o) S

Figure 5.19: The likelihood of proposals being a hand indicated by border color. The blue color indicates unlikely, the red color indicates likely, white lies in between. Colors are derived directly from the cue itself without SVM scoring. Proposals for which the cue was zero or negative are omitted. The VJ cue is omitted because it typically yielded no results.

5.9 Blind Validation

The final validation step evaluates the pipeline with a training and testing set which contains paintings not used during experimentation. The experimentation dataset results were observed and used to adjust algorithmic parameters, which may have lead to overfitting of the pipeline. This section discusses the final results of the four selected painting categories and a mixed set which more closely represents the experimentation dataset by including paintings from all styles.

Table 5.6 shows the performance of the low-level SVM on its own. Aside from Tenebrism, the performance by any metric is about half that of the experimentation dataset, suggesting that indeed overfitting took place. The performance outlier is Tenebrism, which shows a notable improvement in AP and precision, with similar recall. To confirm whether this is because Tenebrism paintings are all similar, i.e, the training set and test have similar hand color/shape characteristics, the test was repeated using the same test data, but using training data from all painting categories. This resulted in 0.118 AP, 5.19% precision and 28.67%, which is a drop, but still notably higher than the experimentation results. This suggests that tenebrism hands are easier to detect, regardless of training data. Tenebrism paintings often contain a single source of light amidst a dark surrounding, which could make it easy to pick up the typically bright colored skin. The mixed category scores better than three of the painting styles, this is not necessarily expected, under the assumption that hands in paintings from the same style are more similar than hands from mixed styles. One way to explain this is that the tuned parameters are optimized for mixed data, rather than data from the same painting category. Academicism performed the worst, which is surprising given the realistic nature of those paintings, looking at the Selective Search performance for the Academicism test set (Table 5.20), only a 71% recall is achieved which could explain the low performance, conversely, Tenebrism has a 93% recall and performed better with the low-level SVM.

	Using Testing Data			*Using Training Data*		
Parameters	**AP**	**P**	**R**	**AP**	**P**	**R**
Tenebrism	0.164	6.82%	35.83%	0.997	99.78%	99.33%
Academicism	0.045	1.08%	14.84%	0.997	100.00%	100.00%
Early Renaissance	0.049	1.42%	14.94%	0.995	99.60%	100.00%
High Renaissance	0.055	1.81%	15.29%	0.997	95.33%	96.00%
Mixed	0.073	2.21%	22.23%	0.990	97.33%	98.00%
Experimentation	0.103	3.18%	38.60%	-	-	-

Table 5.6: SVM results using the evaluation dataset. The last row is from the experimentation dataset and is included for reference.

The experimentation dataset sees a performance improvement when other cues where combined alongside the low-level SVM. The same trend exists

Preset	**Avg. Jaccard**	**Recall**	**Proposals**
Tenebrism	0.730	93%	5,452
Academicism	0.672	71%	5,362
Early Renaissance	0.696	77%	5,962
High Renaissance	0.719	78%	6,214
Mixed	0.741	85%	5,358

Figure 5.20: Selective Search statistics per category. The recall column indicates the percentage of hands that have a proposal with a ground truth Jaccard index of at least 0.5, the avg. Jaccard column is also derived using this threshold.

with the final evaluation datasets. Table 5.7 lists the results of the LSVM weights, and performance metrics. Academicism takes the lead in terms of performance, with 0.25 AP, 12% precision and 49% recall. This is a major gain over using just the low-level SVM, most of this is attributed to DeeperCut. When DeeperCut is used as the only cue, the AP is 0.230 with 10.37% precision and 39.92% recall. A similar trend exists across all styles, where DeeperCut is the main positive performance contributor, and the other cues add little. The effect of DeeperCut is particularly noticeable with Early and High Renaissance, where DeeperCut as a cue on its own performs better than when combined with other cues, in terms of precision but not for recall and AP, i.e., using DeeperCut leads to reduced false-positives but also fewer true-positives.

The performance of Tenebrism, Academicism and mixed all exceed the performance of the experimentation dataset in terms of precision and AP. This trend was not visible with the low-level cues, where each style performed worse than the experimentation set.

Figure A.10 shows some of the hands there were successfully found, and Figure and A.11 shows some typical false-positives.

Style	*LSVM weights*						*performance*		
	DC	**DF**	**S**	**VJ**	**G**	**LL**	**AP**	**P**	**R**
Tenebr.	0.19	0.16	-0.05	0.03	-0.03	0.40	0.230	10.16%	41.47%
Academ.	0.30	0.20	-0.04	0.17	-0.04	0.35	0.250	12.06%	49.03%
Early R.	0.25	0.16	-0.04	0.05	-0.01	0.36	0.174	6.61%	40.38%
High R.	0.18	0.13	-0.03	0.05	-0.04	0.36	0.169	6.55%	36.47%
Mixed	0.22	0.19	-0.00	0.02	-0.01	0.34	0.239	8.96%	52.52%
Exper.	0.21	0.11	-0.09	0.06	-0.08	0.39	0.202	7.13%	37.82%

Table 5.7: LSVM results using the evaluation dataset. The last row is from the experimentation dataset and is included for reference

Chapter 6

Conclusion

This thesis explored the applicability of several cues for the purpose of detecting hands. The low-level cues based on color and shape were encoded using histograms, raw pixels and histograms of oriented gradients (HOG). The encoded data was used to train a support vector machine (SVM) with a radial kernel. The SVM was then employed to classify proposals generated by Selective Search. The resulting proposals, classified as a hand by the SVM, show moderate recall but include very many false-positives. To improve the performance, the 500 best SVM scoring proposals per painting are further classified by combining multiple cues. This culling phase uses five additional cues to encode proposals. 1) DeeperCut human pose detector is used to detect hands and the distance between a proposals and DeeperCut hand is used as cue. 2) Prior information is used to describe common hand locations within a painting as well as 3) Common hand size information. The remaining two cues are derived by firstly detecting faces and then determining the likely hand locations relative to a face. Faces were detected using DeeperCut and Viola and Jones [56] cascading framework for object detection. All the six cues are combined as a feature vector, which is classified by a linear SVM. The first low-level phase on average results in 0.078 AP, 2.67% precision and 21% recall, adding the additional cues these figures improve to 0.21 AP, 7.28% precision and 44% recall. Investigation of the LSVM weights assigned to each of the individual cues shows that the low-level SVM and DeeperCut cues are the important contributors. Applying those two cues mutually exclusively shows that DeeperCut accounts for the majority of positive performance, with the low-level SVM adding more hands, but also more false-positives.

It is shown that detecting hands in certain styles, such as Tenebrism and Academicism is easier than for Early and High Renaissance. DeeperCut is particularly affective with the Academicism style, which is attributed to DeeperCut being pre-trained on photos and Academicism including paintings with high levels of realism in terms of composition, thus Academicism more closely match the training data used by DeeperCut and other styles. The least contributing cue was found to be the one based on the cascading face detector, mostly because it proposes very few actual faces. It is shown that all of the cues contribute to improving the results one way or another, this was confirmed by repeating the experiments with one cue omitted each time.

6.1 Future Work and Improvements

The pipeline shows several areas for improvement. At the basis sits Selective Search, if a hand is not contained within a Selective Search proposed region, it can never be detected by successive components. The chosen Selective Search parameters immediately limit the pipeline performance to approximately 77% recall at best, but with the advantage of having to consider fewer than 5,500 proposals per painting on average. None of the pipeline components approach this figure, suggestion that recall on it's own is not a bottleneck. However, when looking at the Jaccard index when ground truth is matched against Selective Search proposals subject to a 0.5 threshold, the average index is just 0.71. This indicates that when hands are contained inside a Selective Search proposal, they are not perfectly aligned. As a consequence, the descriptor and classifier components of the pipeline never quite see a tightly aligned hand and instead will have to deal with increased amounts of background data that surrounds a hand. Moreover, if hands are not contained tightly within a proposal, the number of ways a stereotypical hand can be expressed is increased, thus demanding the classifiers to learn more permutations of how a hand looks. Future work could focus on making amendments to the Selective Search heuristics to allow a tighter fit of proposals around to be detected objects. Or alternatively, the generic version of Selectie Search could be kept, but proposals could be normalized in a post-processing phase to better align with prior data (e.g., shift the proposal such that the majority of skin colored pixels are near the center, as would be expected of a hand).

From the low-level cues used to train the SVM it is unexpected that raw pixels performs the best, notably because this approach is rarely used by other researchers for object detection in images. Moreover, it is unexpected that the the 4 by 4 pixel sized descriptor performs the best. This configuration does not allow a great number of degrees to describe hands, and effectively uses an average color taken from 16 quadrants of a proposal. The key reason it is included is because it performed better than color histograms in practise. The poor color histogram performance is attributed to the high variance in colors and possibly the imbalance between positive and negative training data. Although the descriptors are mean and variance normalized, this is done per single dimension acros all training data, improvements might be achieved by normalizing the paintings colors prior to extracting the descriptors. Future work may also employ an approach that does not use uniformly spaced bins, but rather spaces the bins based on prior data, e.g., more skin colored bins and fewer background colored bins. The HOG descriptor is not typically used to describe objects with high degrees of freedom, nor is it used to detect smaller objects, it is not expected that the descriptor can be significantly improved upon for the purpose of detecting hands in Renaissance era paintings. In resent work Roy et al. [43] entirely forego the explicit encoding and aggregating of colors and use a CNN to automatically do this. In their work two CNNs are combined to 1) detect skin regions among Selective Search proposals 2) from the skin regions detect hands. In general, with the advent of computer processing power, CNN approaches frequently outperform SVM approaches. Future work should determine how Roy et al. [43] their pipeline performs on Renaissance era paintings.

DeeperCut is a key contributor in the pipeline to find hands. Because DeeperCut comes with a pre-trained CNN, it is expected that through transfer learning the performance can be improved. This transfer learning approach was successfully applied by Westlake et al. [57] for the purpose of detecting people in paintings. Moreover, the input images for DeeperCut were resized such that no dimension exceeds 512 pixels. This was necessitated due to hardware constraints, it is not researched whether this had any impact on the performance in terms of precision and recall.

The use of hand size and spatial priors could be improved by gathering more training data. Data in the pipeline only used information from 50 paintings, and the use convolution was needed to add nuisances. Although experiments used varying kernel sizes, it is not well understood how the number paintings influenced the results. Convolution may have also added an unfair bias because quite a few paintings include Madonna holding baby Jesus, which quickly adds 4 hands within close proximity. The hand size prior could also be encoded as Selective Search heuristic to allow more hand sized proposals, and cull unlikely proposals prior to reaching the SVM.

Moreover, some concessions in the pipeline were made due to limited available processing power. The paintings where shrunk to be less than 600 pixels in width and height, which may have lead to a loss of detail, especially considering that hands are typically quite small. Likewise, this also necessitated the need for the Selective Search *fast* configuration. It would be interesting to see how well the *quality* configuration performs, because its average Jaccard index is higher.

Bibliography

[1] Vassilis Athitsos and Stan Sclaroff. Estimating 3d hand pose from a cluttered image. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–432. IEEE, 2003.

[2] Sven Bambach, Stefan Lee, David J Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1949–1957, 2015.

[3] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):509–522, 2002.

[4] Christopher M Bishop. *Pattern recognition and machine learning.* Springer, 2006.

[5] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 8(6):679–698, 1986.

[6] Olivier Chapelle, Patrick Haffner, and Vladimir N Vapnik. Support vector machines for histogram-based image classification. *IEEE transactions on Neural Networks*, 10(5):1055–1064, 1999.

[7] Neil Collins. Visual Arts Encyclopedia. `http:/www.visual-arts-cork.com/`, 2016.

[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[9] Nasser H Dardas and Nicolas D Georganas. Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. *IEEE Transactions on Instrumentation and Measurement*, 60(11):3592–3607, 2011.

[10] Wang Dong and Zhou Shisheng. Color image recognition method based on the prewitt operator. In *Computer Science and Software Engineering, 2008 International Conference on*, volume 6, pages 170–173. IEEE, 2008.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[13] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2): 167–181, 2004.

[14] Pedro F Felzenszwalb, Ross B Girshick, D McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2009. ISSN 0162-8828.

[15] Jannik Fritsch, Sebastian Lang, A Kleinehagenbrock, Gernot A Fink, and Gerhard Sagerer. Improving adaptive skin color segmentation by incorporating results from face detection. In *Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on*, pages 337–343. IEEE, 2002.

[16] Shiry Ginosar, Daniel Haas, Timothy Brown, and Jitendra Malik. Detecting people in cubist art. In *Workshop at the European Conference on Computer Vision*, pages 101–116. Springer, 2014.

[17] Giovani Gomez and Eduardo F. Morales. Automatic feature construction and a simple rule induction algorithm for skin detection. *Proc. of the ICML workshop on Machine Learning in Computer Vision*, pages 31–38, 2002.

[18] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, and Gang Wang. Recent advances in convolutional neural networks. *arXiv preprint arXiv:1512.07108*, 2015.

[19] Steve R Gunn. On the discrete representation of the laplacian of gaussian. *Pattern Recognition*, 32(8):1463–1472, 1999.

[20] Nariman Habili, Cheng Chew Lim, and Alireza Moini. Segmentation of the face and hands in sign language video sequences using color and motion cues. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(8):1086–1097, 2004.

[21] Abdenour Hadid, Matti Pietikainen, and Timo Ahonen. A discriminative feature space for detecting and recognizing faces. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2004.

[22] Peter Hall, Hongping Cai, Qi Wu, and Tadeo Corradi. Cross-depiction problem: Recognition and synthesis of photographs and artwork. *Computational Visual Media*, 1(2):91–103, 2015. ISSN 2096-0433.

[23] Chen Chiung Hsieh, Dung Hua Liou, and Wei Ru Lai. Enhanced face-based adaptive skin color model. *Journal of Applied Science and Engineering*, 15(2):167–176, 2012. ISSN 15606686.

[24] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification. *Tech. rep., Department of Computer Science, National Taiwan University.*, 2003.

[25] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model. *Arxiv*, 2016. ISSN 0302-9743.

[26] Itseez. Open source computer vision library. `https://github.com/itseez/opencv`, 2017.

[27] Oliver Jesorsky, Klaus J Kirchberg, and Robert W Frischholz. Robust face detection using the hausdorff distance. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 90–95. Springer, 2001.

[28] Michael J Jones and James M Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[29] Praveen Kakumanu, Sokratis Makrogiannis, and Nikolaos Bourbakis. A survey of skin-color modeling and detection methods. *Pattern recognition*, 40(3):1106–1122, 2007.

[30] Iasonas Kokkinos, Michael Bronstein, and Alan Yuille. Dense scale invariant descriptors for images and surfaces. Technical Report RR-7914, hal-00682775, INRIA, 2012.

[31] Mathias Kolsch and Matthew Turk. Analysis of rotational robustness of hand detection with a viola-jones detector. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 107–110. IEEE, 2004.

[32] Mathias Kölsch and Matthew Turk. Robust hand detection. In *FGR*, pages 614–619, 2004.

[33] Jure Kovac, Peter Peer, and Franc Solina. *Human skin color clustering for face detection*, volume 2. IEEE, 2003.

[34] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[35] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. Evaluation of ranked retrieval results. In *Introduction to information retrieval*, chapter 8.4, pages 158–164. Cambridge university press, 2008.

[36] Arpit Mittal, Andrew Zisserman, and Philip HS Torr. Hand detection using multiple proposals. In *BMVC*, pages 1–11. Citeseer, 2011.

[37] Trong-Nguyen Nguyen, Huu-Hung Huynh, and Jean Meunier. Static hand gesture recognition using principal component analysis combined with artificial neural network. *Jounal of Automation and Control Engineering*, 3(1):40–45, 2015.

[38] Eng-Jon Ong and Richard Bowden. A boosted classifier tree for hand shape detectiona boosted classifier tree for hand shape detection. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 889–894. IEEE, 2004.

[39] Giorgio Panin, Sebastian Klose, and Alois Knoll. Real-time articulated hand detection and pose estimation. In *International Symposium on Visual Computing*, pages 1131–1140. Springer, 2009.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[41] Son Lam Phung, Abdesselam Bouzerdoum, and Douglas Chai. Skin segmentation using color and edge information. In *Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on*, volume 1, pages 525–528. IEEE, 2003.

[42] Son Lam Phung, Abdesselam Bouzerdoum, and Douglas Chai. Skin segmentation using color pixel classification: analysis and comparison.

IEEE transactions on pattern analysis and machine intelligence, 27(1): 148–154, 2005.

[43] Kankana Roy, Aparna Mohanty, and Rajiv R Sahay. Deep learning based hand detection in cluttered environment using skin segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 640–649, 2017.

[44] Pierre-André Savalle, Stavros Tsogkas, George Papandreou, and Iasonas Kokkinos. Deformable part models with cnn features. In *European Conference on Computer Vision, Parts and Attributes Workshop*, 2014.

[45] Joseph Schlecht, Bernd Carqué, and Björn Ommer. Detecting gestures in medieval images. In *2011 18th IEEE International Conference on Image Processing*, pages 1285–1288. IEEE, 2011.

[46] Karin Sobottka and Ioannis Pitas. A novel method for automatic face segmentation, facial feature extraction and tracking. *Signal processing: Image communication*, 12(3):263–281, 1998.

[47] Bjoern Stenger, Arasanathan Thayananthan, Philip HS Torr, and Roberto Cipolla. Hand pose estimation using hierarchical detection. In *International Workshop on Computer Vision in Human-Computer Interaction*, pages 105–116. Springer, 2004.

[48] Björn Stenger. Template-based hand pose recognition using multiple cues. In *Asian Conference on Computer Vision*, pages 551–560. Springer, 2006.

[49] Ekaterini Stergiopoulou, Kyriakos Sgouropoulos, Nikos Nikolaou, Nikos Papamarkos, and Nikos Mitianoudis. Real time hand detection in a complex background. *Engineering Applications of Artificial Intelligence*, 35:54–70, 2014.

[50] Feng Tang, Shane Brennan, Qi Zhao, and Hai Tao. Co-tracking using semi-supervised support vector machines. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[51] Michael J Taylor and Tim Morris. Adaptive skin segmentation via feature-based face detection. In *SPIE Photonics Europe*, pages 91390P–91390P. International Society for Optics and Photonics, 2014.

[52] Arasanathan Thayananthan, Bjoern Stenger, Philip HS Torr, and Roberto Cipolla. Shape context and chamfer matching in cluttered

scenes. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2003.

[53] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[54] Remco C Veltkamp and Michiel Hagedoorn. State of the art in shape matching. In *Principles of visual information retrieval*, pages 87–119. Springer, 2001.

[55] Vladimir Vezhnevets. A Survey on Pixel-Based Skin Color Detection Techniques. *Cybernetics*, 85(0896-6273 SB - IM):85–92, 2003. ISSN 08966273.

[56] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[57] Nicholas Westlake, Hongping Cai, and Peter Hall. Detecting people in artwork with cnns. In *European Conference on Computer Vision*, pages 825–841. Springer, 2016.

[58] WikiArt. WikiArt.org - Visual Art Encyclopedia. `http:/wikiart.org/`, 2016.

[59] Qi Wu and Peter Hall. Modelling visual objects invariant to depictive style. In *BMVC*, 2013.

[60] Qi Wu, Hongping Cai, and Peter Hall. Learning graphs to model visual objects across different depictive styles. In *European Conference on Computer Vision*, pages 313–328. Springer, 2014.

[61] Ying Wu and Thomas S Huang. Vision-based gesture recognition: A review. In *Gesture Workshop*, volume 1739, pages 103–115. Springer, 1999.

[62] Bangpeng Yao and Li Fei-Fei. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (9):1691–1703, 2012. ISSN 01628828.

[63] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[64] Xiaojin Zhu, Jie Yang, and Alex Waibel. Segmenting hands of arbitrary color. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 446–453. IEEE, 2000.

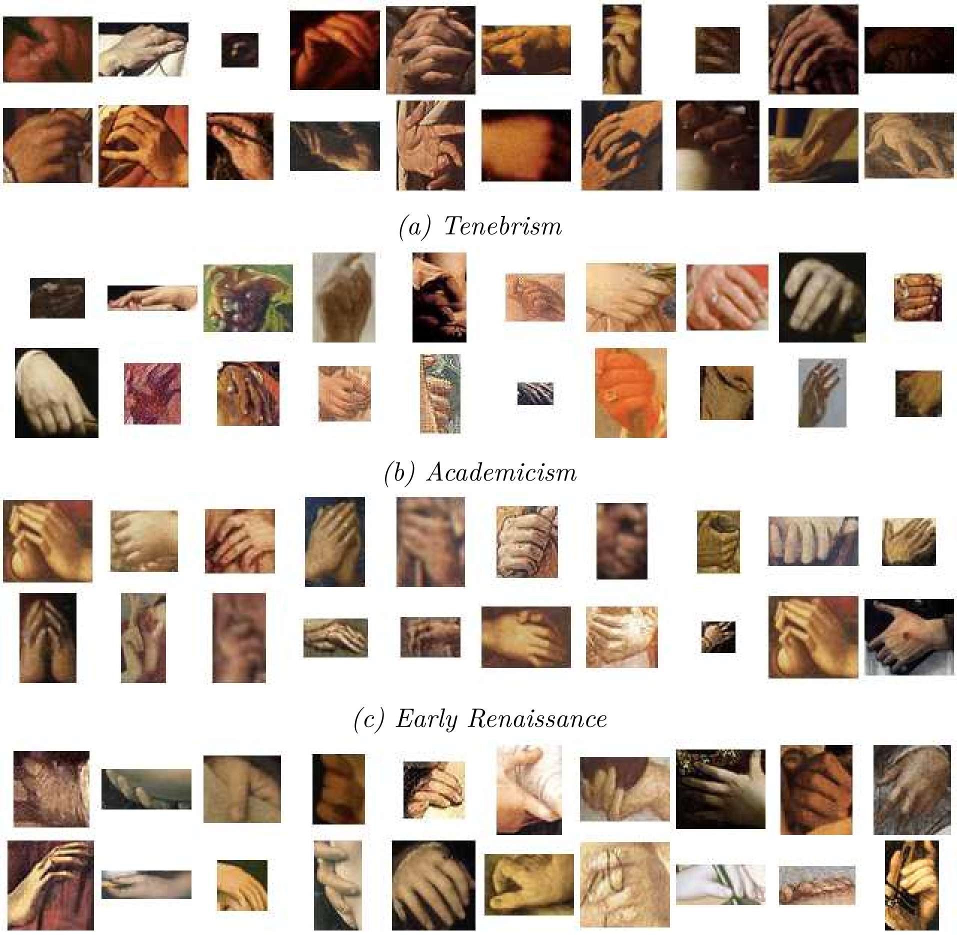
[65] Jorn Alexander Zondag, Tommaso Gritti, and Vincent Jeanne. Practical study on real-time hand detection. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, pages 1–8. IEEE, 2009.

Appendices

Appendix A

Supporting Figures, Tables and Graphs.

A.1 Example Annotated Hands



(a) Tenebrism

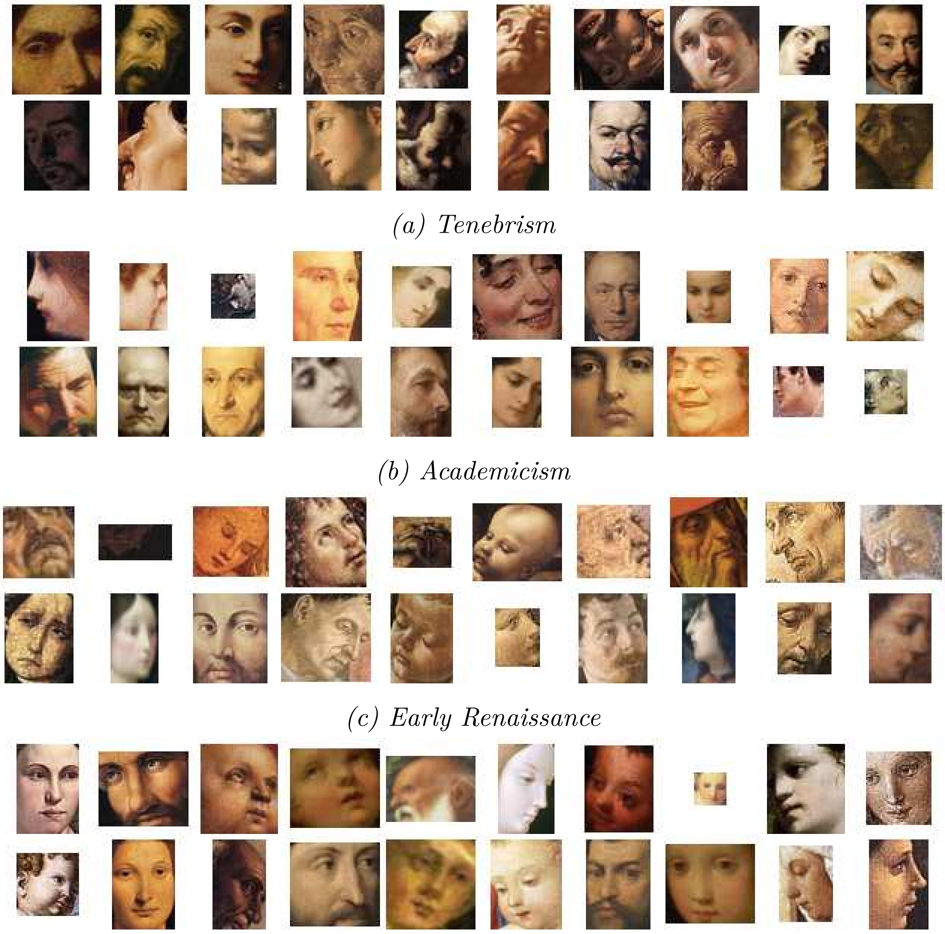
(b) Academicism

(c) Early Renaissance

(d) High Renaissance

Figure A.1: Hands extracted as annotated. Images are resized such that no dimension is greater than 60 pixels, while maintaining aspect ratio.

A.2 Example Annotated Faces



(a) Tenebrism

(b) Academicism

(c) Early Renaissance

(d) High Renaissance

Figure A.2: Faces extracted as annotated. Images are resized such that no dimension is greater than 60 pixels, while maintaining aspect ratio.

A.3 Annotation Tool Screenshot

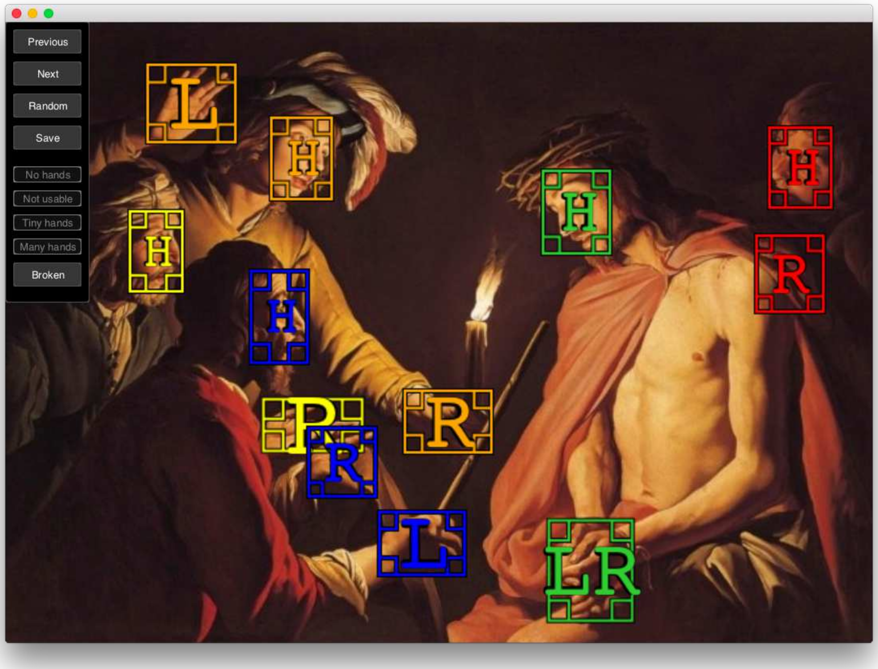
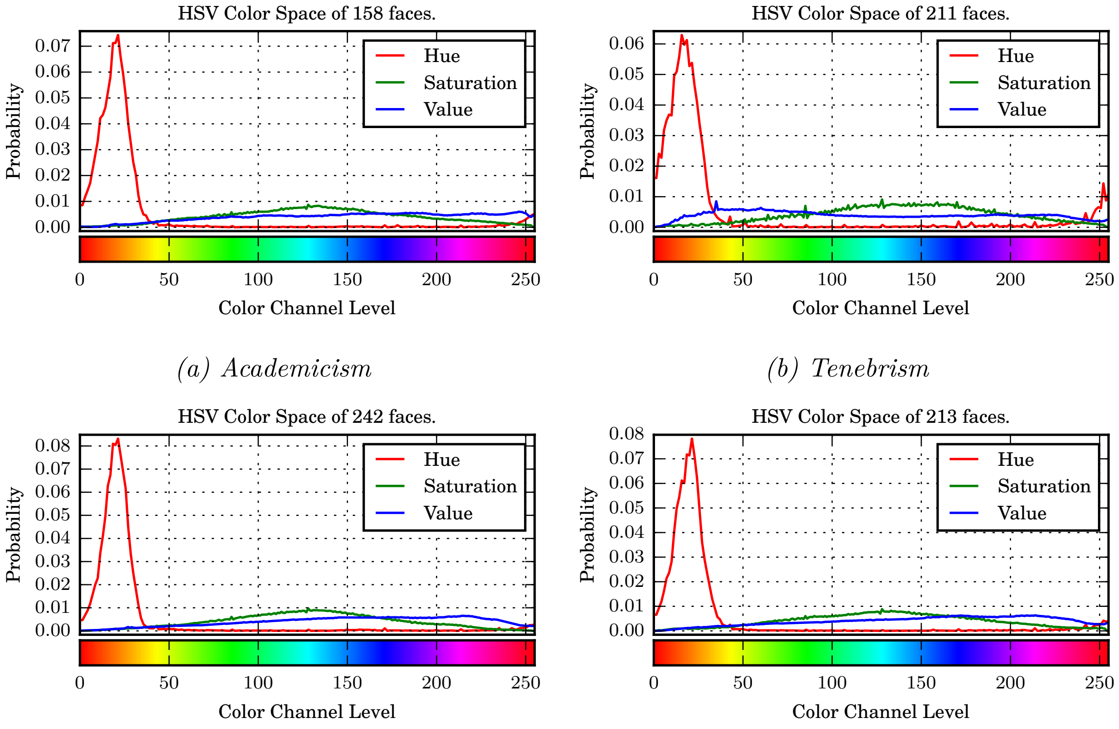


Figure A.3: The annotation tool in use. Each annotated person has a uniquely associated color. NB.: The 'broken' button selects a painting with invalid annotations, such as multiple heads per person.

A.4 Face Color HSV Plots



(a) Academicism

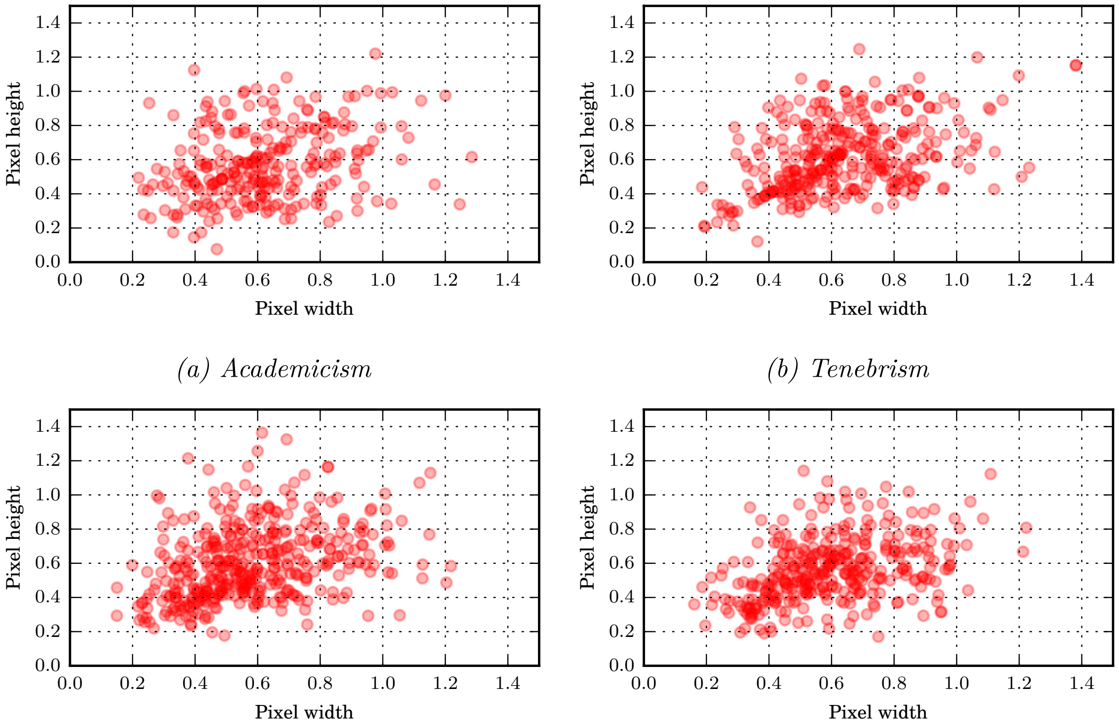
(b) Tenebrism

(c) Early Renaissance

(d) High Renaissance

Figure A.4: HSV histogram plot per annotated face category.

A.5 Hand Sizes Per Category



(a) Academicism

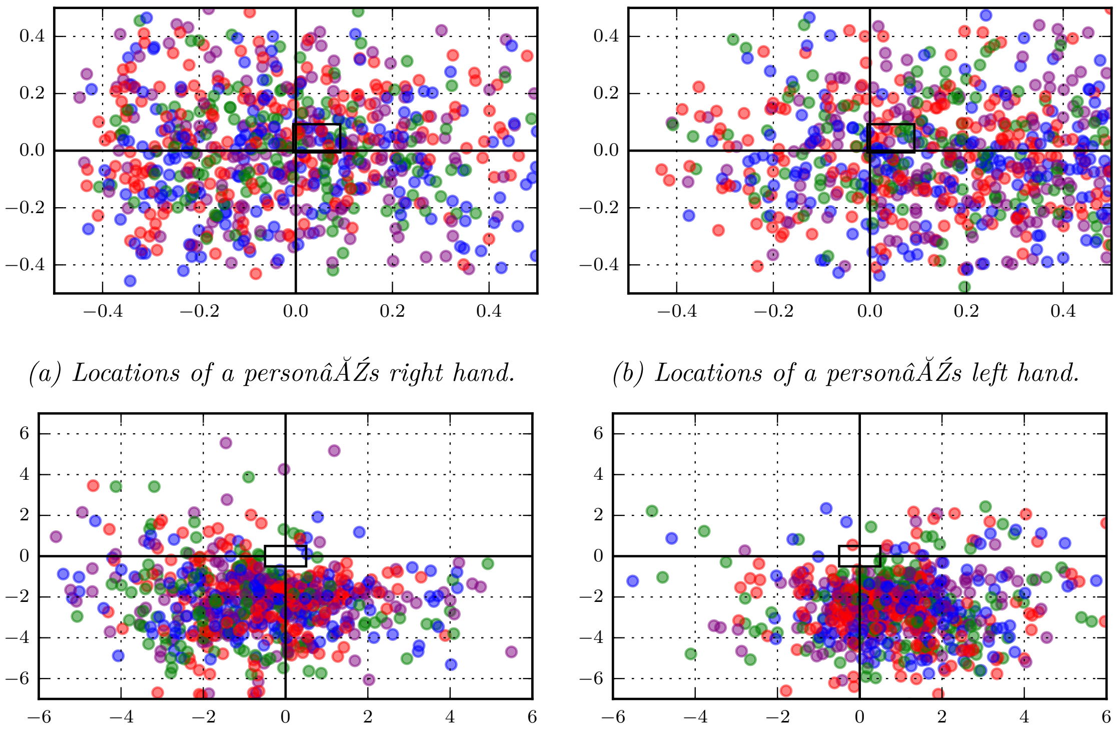
(b) Tenebrism

(c) Early Renaissance

(d) High Renaissance

Figure A.5: Hand sizes normalized by each persons annotated face's diagonal. Assorted per painting category.

A.6 Scatter Priors per Category



(a) Locations of a personâĂŹs right hand.

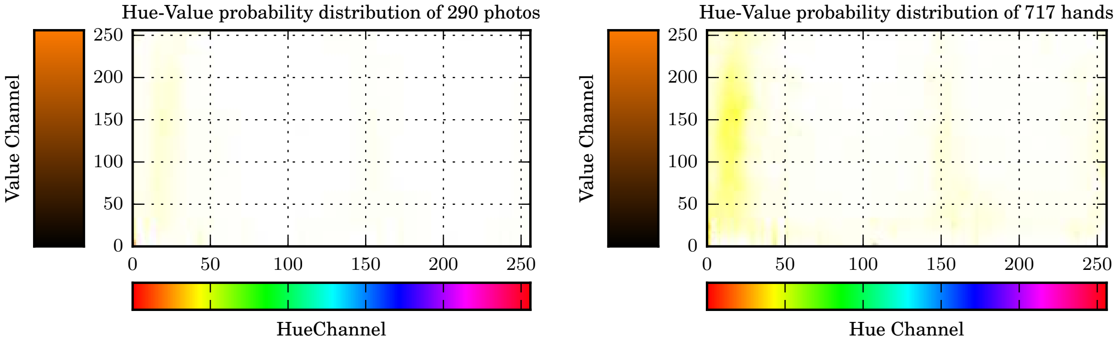
(b) Locations of a personâĂŹs left hand.

(c) Relative right hands.

(d) Relative left hands.

Figure A.6: Hand positions color coded per painting category; purple: Tenebrism, green: Academicism, red: Early Renaissance, blue: High Renaissance. The origin indicates the center of either the painting (A.6a and A.6b) or a head (A.6c and A.6d).

A.7 Hue-Value Correlation in Photos

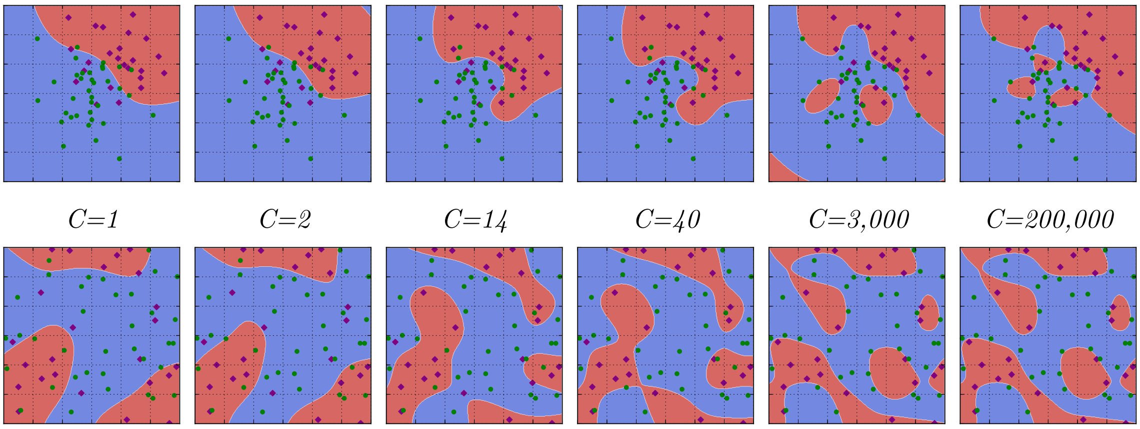


(a) Photos

(b) Photo hands

Figure A.7: Hue-Value correlation plot from photos. Generated from the dataset used by Mittal et al. [36]. Probability runs from red (high) to yellow (medium) to white (never).

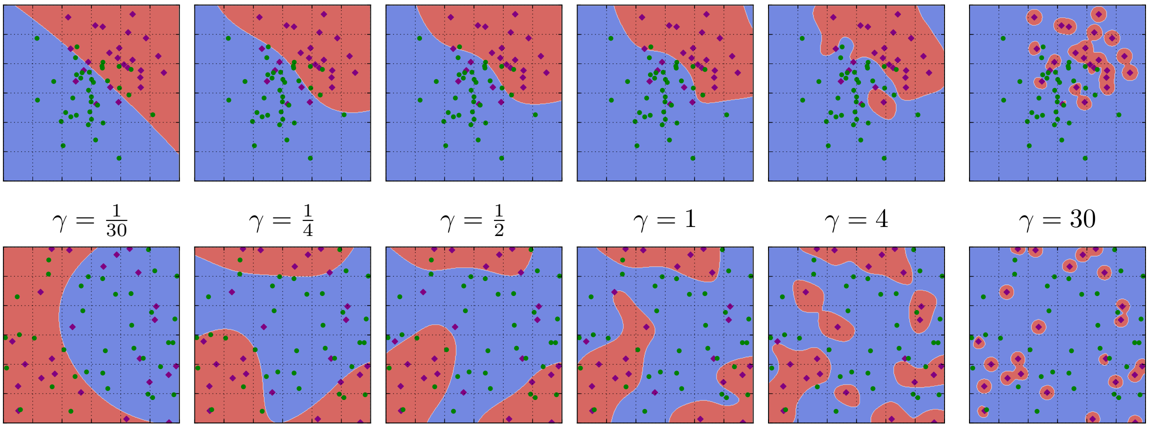
A.8 SVM Radial Kernel Experiments



C=1 *C=2* *C=14* *C=40* *C=3,000* *C=200,000*

C=1 *C=2* *C=14* *C=40* *C=3,000* *C=200,000*

Figure A.8: Decision boundaries for various slack (c) values using a radial kernel (rbf). Points in the top row of figures are sampled from a normal distribution with each class having a different mean. Points in the bottom row of Figures are sampled from a uniform distribution. Gamma (γ) is set to $\frac{1}{2}$.



$\gamma = \frac{1}{30}$ $\gamma = \frac{1}{4}$ $\gamma = \frac{1}{2}$ $\gamma = 1$ $\gamma = 4$ $\gamma = 30$

$\gamma = \frac{1}{30}$ $\gamma = \frac{1}{4}$ $\gamma = \frac{1}{2}$ $\gamma = 1$ $\gamma = 4$ $\gamma = 30$

Figure A.9: Decision boundaries for various gamma (γ) values using a radial kernel (rbf). Points in the top row of figures are sampled from a normal distribution with each class having a different mean. Points in the bottom row of Figures are sampled from a uniform distribution. The slack value (c) is set to 1.

A.9 Examples of True Positives

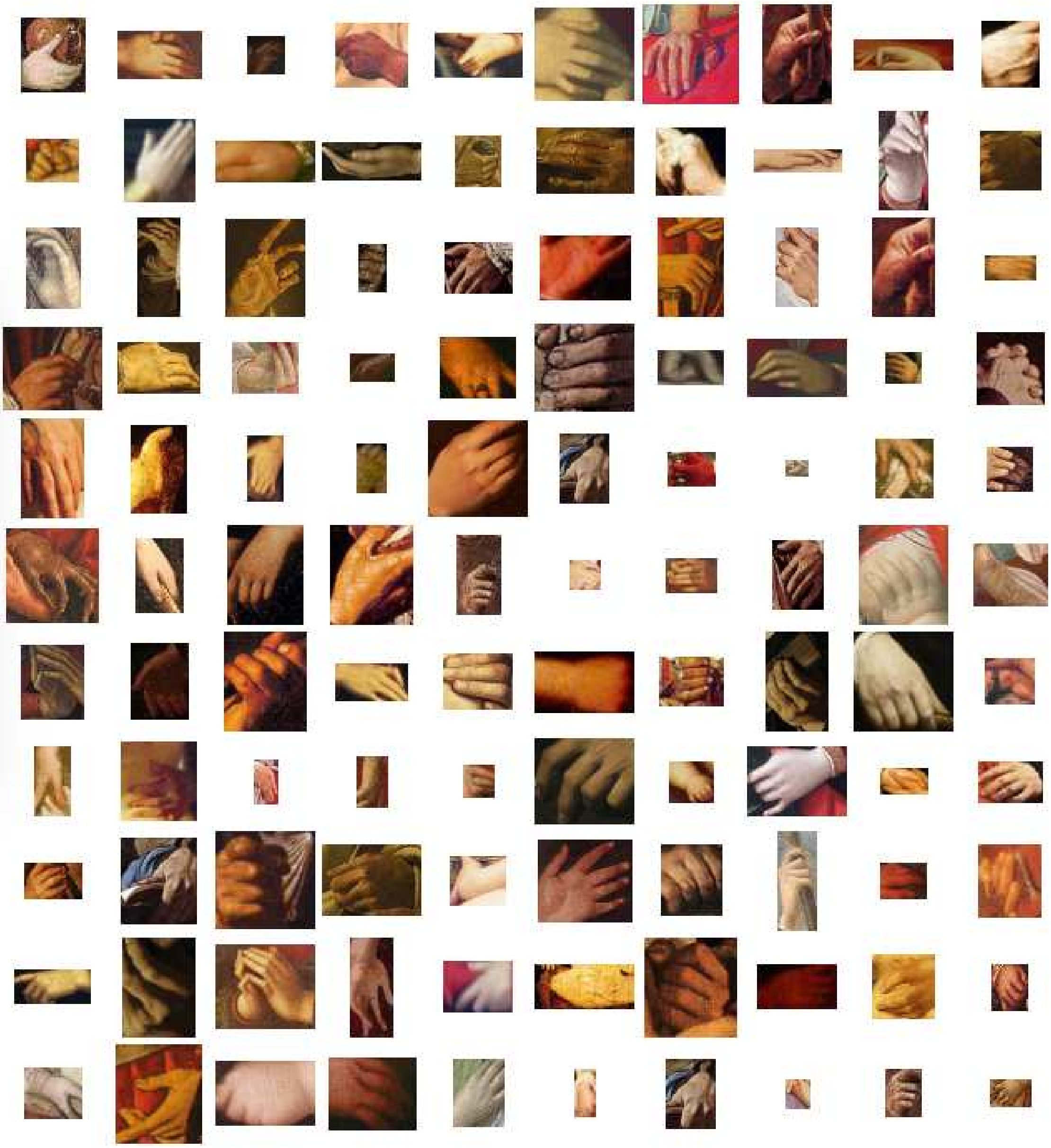


Figure A.10: Some of the hands found by the pipeline in the final dataset. Images taken from across all styles.

A.10 Examples of False Positives

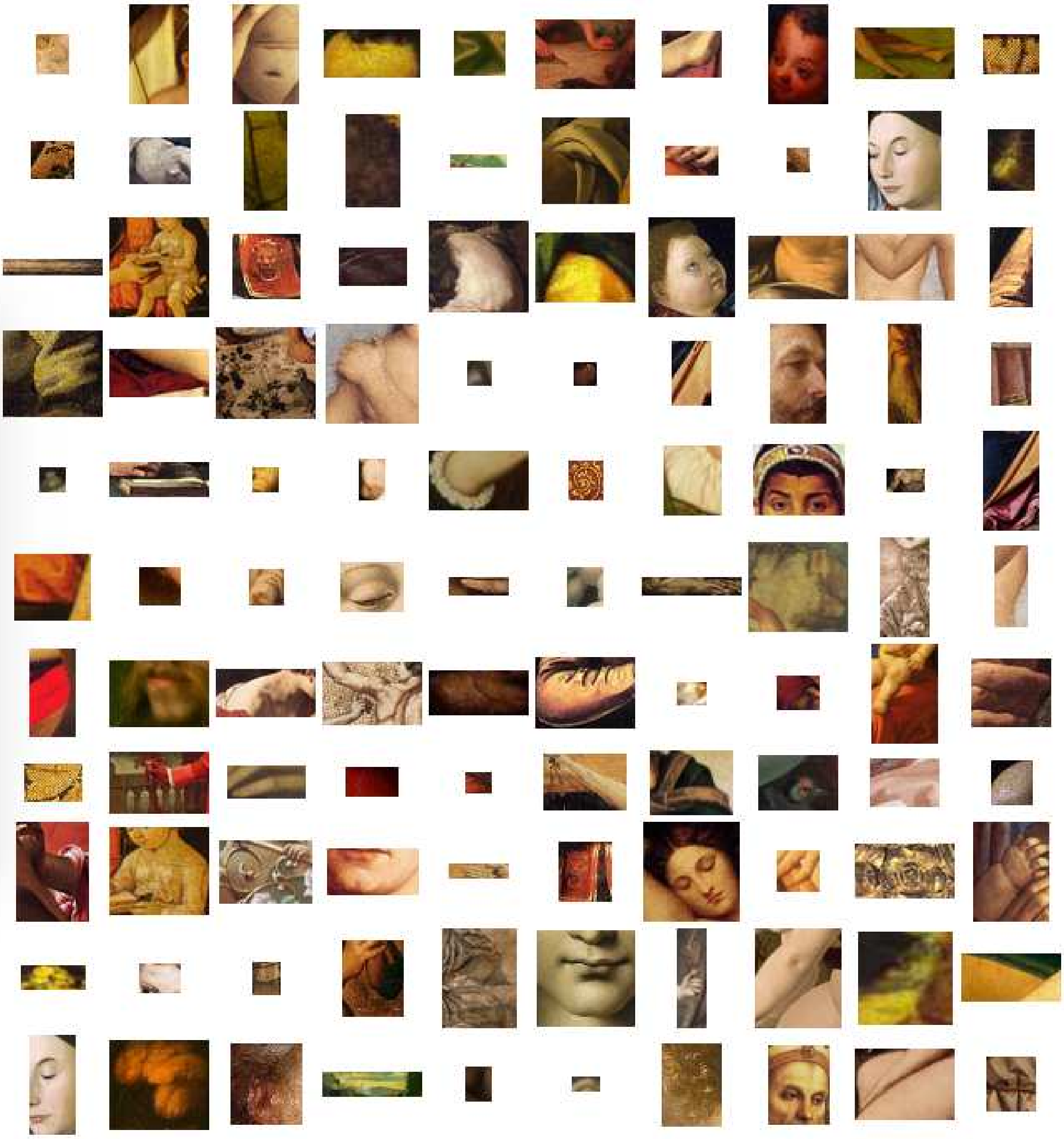
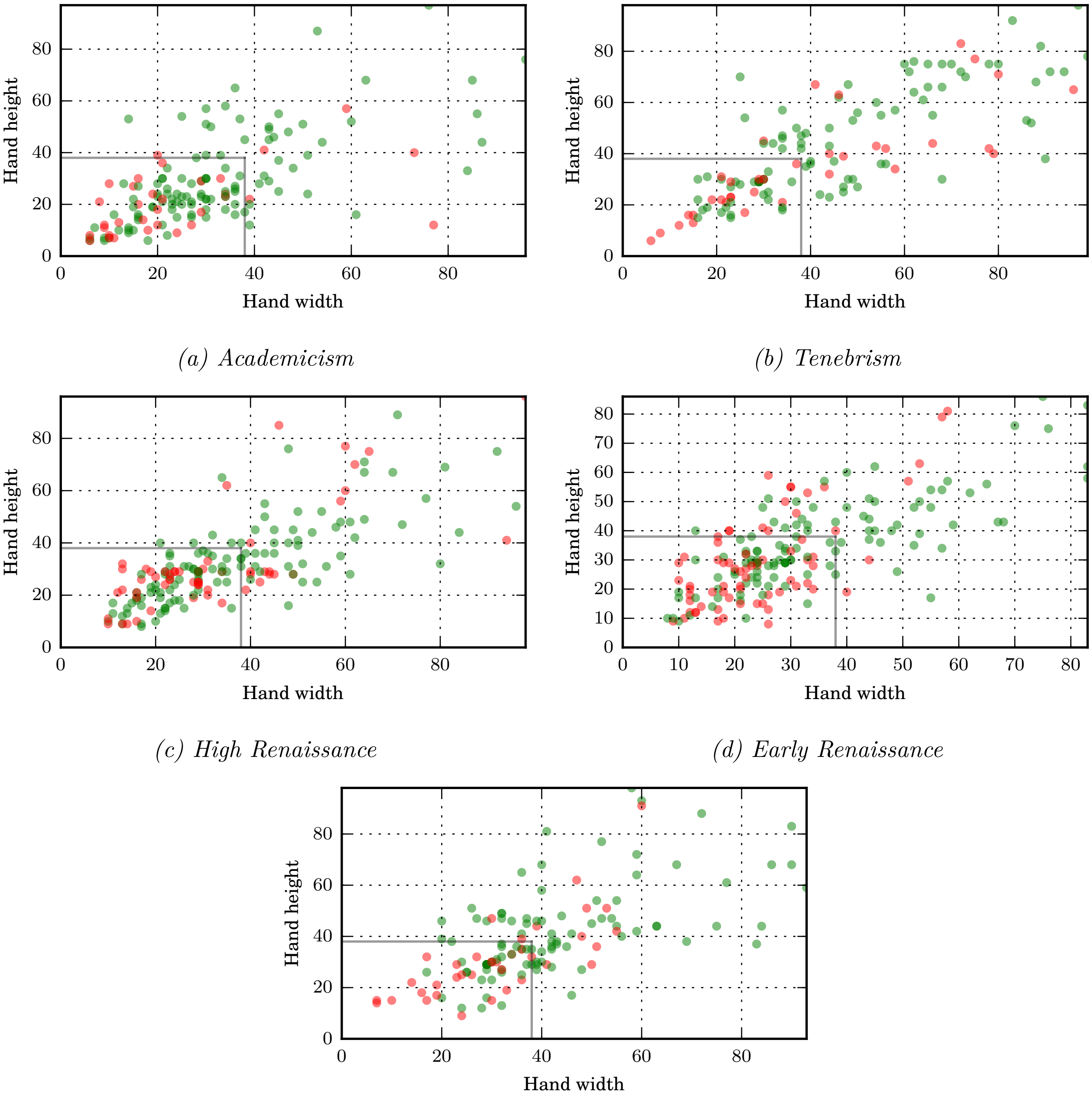


Figure A.11: False-positive hands as found by the pipeline in the final dataset. Images taken from across all styles.

A.11 Hand Sizes per Style



(a) Academicism

(b) Tenebrism

(c) High Renaissance

(d) Early Renaissance

(e) All

Figure A.12: Correlation between true-positives (green) and false-negatives (red). Some outliers are removed for visual convenience. The sliding window size of 38 *by* 38 *pixels in marked with a light gray line.*