Forgery Detection in Paintings Using North Carolina Bird Series

Bridget Willke

Duke University

April 14, 2016

Abstract

This report investigates a combination of several techniques in machine learning, mathematical modeling and image processing, which are used to detect forgeries in pairs of unauthenticated digitized paintings. These methods are implemented without initial input from art experts, and therefore could potentially be used as a tool to aid in the validation of unidentified paintings in the future. In this report, a series of paintings (originals and copies), commissioned for scholarly analysis of forgery detection, are analyzed to test classification models. The data is extracted from each scanned painting using Dual-Tree Wavelet transforms, producing features for different patches of the painting. These features and the underlying structure of the image are captured using a Hidden Markov Tree model. Support Vector Machines with different kernel functions, following methods from a previous experiment by a Princeton team (Polatkan et al., 2009), were used to perform classification. After analyzing and improving the models, different pairs of paintings were tested. While classification of paintings (originals or copies) was generally accurate when using patches from the same paintings in both the testing and training sets, it was less successful when the paintings used in testing and training sets were different.

Keywords: forgeries, wavelets, hidden markov-tree models, support vector machines

Introduction

The algorithms implemented in this report combine applied mathematics, statistics, computer science, art history and image processing (Sapiro, 2014; Gonzalez and Woods, 2002). Using a new data set, the North Carolina bird series, we seek to replicate and improve the results obtained by a Princeton team a few years ago (Polatkan et al., 2009). Part 1 of this report replicates prior research and investigates the effects of image registration on forgery detection results. Part 2 uses a new data set and improves the models used in prior reports using variable selection. Part 3 extends the scope of Part 2, using one or more pairs of paintings to predict the identity of a second pair of paintings. All analyses in this paper were executed in MATLAB, unless otherwise stated (MATLAB, 2015).

Data Sets

Museums have made several data sets available over the past few years to research of forgery detection in paintings. To allow a more scholarly analysis, Charlotte Caspers, an artist specialized in art reconstruction, was invited to paint a series of seven small paintings (of size about 25 cm \times 20 cm) in 2008, in order to study the differences between originals and copies (Wang, 2015). These paintings were carried out in a variety of styles and with different materials. For each of the seven paintings, Caspers also painted a copy of her own, typically a few days after she painted the original. Every copy was painted with the same paints and brushes, on identical ground and under the same lighting (Wang, 2015). The pairs of originals and copies were digitized on a high-resolution scanner for analysis purposes. The task is to build and train a mathematical model to distinguish copies from originals in this data set.

Limitations of 2008 Data Set

This 2008 data set was studied by two groups of mathematicians, the Princeton University group and the France group (Abry et al., 2013; Polatkan et al., 2009). The two groups applied different methods, but for the purpose of this report we will focus on the work done by the Princeton group. The 2008 data set (Caspers, 2008) is limited by the number of pairs of originals and copies, as well as the high variation of painting conditions (materials, paint brush, painting content) across different pairs of paintings. Therefore, it's hard to verify accuracy of a model trained on some pairs by testing on the remaining pairs. In addition, the pairs painted on canvas pose an extra complexity, as the canvas pattern appears in the digitized painting and it's unclear whether the model picks up canvas pattern or brush stroke pattern. In a separate study reviewing the Princeton group's work, it was shown that digitally removing the canvas pattern and training the model on "de-canvassed" images caused the classification accuracy drop, compared to the model trained directly on images with canvas pattern (Wang, 2015).

2012 Data Set

Charlotte Caspers was invited to Duke in 2012 in order to provide researchers with a new, more consistent data set. During her stay, she painted a collection of paintings - the North Carolina bird series, containing 8 pairs of originals and copies (Caspers, 2012). For two of these pairs, we know which is the original and which is the copy. For the other pairs,

Caspers also marked them on the back so certification of copy/original is possible from her records. These records have been kept secret from us, and the identity of copy/original is therefore not known by the mathematical researchers. All the paintings in this new data set are painted on board using consistent materials and techniques. Therefore, it's possible to carry out a more systematic study of the previous methods applied to the old painting set by experimenting on the new collection. In addition, two other local artists, Whitney McCray and Jan Dickey were invited to paint copies of selected paintings from Caspers's collection. It's thus possible to quantify differences in paintings by different painters when studying the new data set. For the purpose of this analysis, we will focus on the copies painted by Dickey.

1 PART 1

Expanding previous work by analyzing original images, forgeries, and a registered version of the forgery onto the original image.

1.1 Methods

1.1.1 Sample Data Selection

For the initial stages of this project, one pair of paintings on wood board from the 2008 data set (Caspers, 2008) was selected as a testing and learning data set. The selection of this testing pair addressed confounding factors present in previous analyses: image quality and the material on which the painting was created (canvas, wood board). The selected paintings were scanned in a consistent manner, and were both painted on wood board, a surface shown in previous analyses to introduce less bias than rough canvas (Polatkan et al., 2009; Wang, 2015).

Figure 1: Paintings from 2008 Data Set by Charlotte Caspers



(a) Charlotte 6



(b) Charlotte 7

Seven patches were selected from corresponding areas in each painting. The paintings will be referred to as "Charlotte 6" and "Charlotte 7", shown in figure 1. These patches were 256×256 pixels, taken from various parts of the bird's body and the pomegranate. The selections were not random, as they had been in previous analyses. Because Caspers did not attempt to replicate the background area when creating the "copy" painting - she only mimicked the styles and aspects of the main subjects of the painting - sampling from the background areas would not be representative of the artist's ability to reproduce the style from the "original".

1.1.2 Extensions and Preprocessing

In the previous analyses, only patches from the original and copy paintings were analyzed and compared. However, after considering ways to expand the analysis, another option was introduced for this current analysis. Since the images are supposed to be very similar, it might yield interesting results to register a patch from the copy onto the corresponding patch from the original, and inspect the resulting registered image, to add an additional dimension to the analysis.

An image-processing program, *Fiji* (Schindelin et al., 2012), was used to implement this image registration. Several different registration plugins were tested, with *"bUnwarpJ"* (Arganda-Carreras et al., 2006) and *"TurboReg"* (Thévenaz et al., 1998), both affine transformations, successfully yielding potentially useful results.

After attempting "*bUnwarpJ*" on several of the patches, the results, although compiled successfully, were often too blurry to be useful. Instead, "*TurboReg*" was selected as the ultimate method for registration; the transformations were affine, but not overly warped to the point of losing information in the image, as was seen in "*bUnwarpJ*".

To implement *"TurboReg"*, a pair of corresponding patches from "Charlotte 6" and "Charlotte 7" (for example, the bird's eye in both images) was imported into Fiji. The images were converted to RGB stack, as necessary for *"TurboReg"*. For this registration method, one channel (R, G or B) is selected to create an initial transformation map. Red was selected for this transform, as it was the most prominent color in our test paintings. After selecting the channel, the copy was mapped onto the original to produce the final registered image.

1.2 Experiment

1.2.1 Feature Vector Extraction Overview

After obtaining the seven patches from the original, the corresponding seven patches from the copy, and the registered versions of these seven areas, the patches are analyzed using the feature extraction code from the previous analysis (Polatkan et al., 2009). During the original analysis, the Princeton group applied image processing and machine learning tools to detect the copies (Polatkan et al., 2009); this method will be implemented for our analysis as well.

1.2.2 Dual Tree Wavelet Transform

In the image processing, a Dual-Tree Complex Wavelet Transform was used (MathWorks, 2016a); this transform is a specific wavelet transform in multi-scale harmonic analysis with better direction selection than normal tensor wavelet transform in 2D. Dual-Tree Wavelet transforms allow better recognition of edges in the patches that were neither horizontal nor vertical, and therefore provide a better representation of the image. In addition, the Dual-Tree Wavelet transform is shift-invariant such that a small shift in the image won't result in big changes in the magnitude of its transformed coefficients.

1.2.3 Hidden Markov Tree Model

After using this wavelet transform, a Hidden Markov Tree (HMT) model was used to capture the underlying statistical structure of the image. The Hidden Markov Tree model is a way to

predict an unknown binary outcome, given certain observations (de Freitas, 2012). Let X represent our hidden state and Y represent our observations. Then the posterior probability of the hidden state X, given the observations can be written as

$$P(X = x_i | Y = y_i) \tag{1}$$

and we can model a prior P(X) and a likelihood, P(Y|X). However, instead of using the same prior for every state, we use an initial prior $P(X_0)$, and then update the prior based on previous states; $P(X_t|X_{t-1})$ serves as a transition prior. Finally, we can compute the posterior using this transition prior and likelihood (de Freitas, 2012). Let $Y_{1:t}$ represent the set $\{Y_1, Y_2, Y_3, ...\}$, then using Bayes' theorem,

$$P(X_t|Y_{1:t}) = \frac{P(Y_t|X_t)(X_t|Y_{1:t-1})}{\sum_{x_t} P(Y_t|X_t)(X_t|Y_{1:t-1})}$$
(2)

In the wavelet decomposition, the unknown state is predicted by observing different wavelet coefficients, as well as the probability of transitioning from one state to another, given the current state. Using this set of priors and observations, a posterior distribution of the coefficients can be calculated to predict the final state. For this data set, there are nodes of the HMT, with two hidden states at each node, which control the magnitude of our wavelet coefficients (Polatkan et al., 2009).

At each level, there are several parameters (Polatkan et al., 2009):

- σ_S : Variance of the narrow Gaussian distribution.
- σ_L : Variance of the wide Gaussian distribution.
- α_T : A 2 × 2 transition probability matrix Pr[child | parent].

$$\begin{array}{|c|c|c|}\hline P_{SS}: Pr(small|small) & 1 - P_{LL}: Pr(small|large) \\\hline 1 - P_{SS}: Pr(large|small) & P_{LL}: Pr(large|large) \\\hline \end{array}$$

The transition matrix describes the probability of having either the small or the large variance parameter, given the size of the parent's variance. The variances are both from Gaussian distributions; one is a narrow Gaussian, which will be referred to as the "small" variance, and one is from a wide Gaussian, which will be referred to as the "large" variance. Using these variances and the transition matrix, we can attempt to predict the unknown variance of the hidden nodes in our model, given the variances we have already observed. In this analysis, 30 pairs of transition probabilities of wavelet coefficients were observed - 5 scales × 6 directions with P_{SS} and P_{LL} for each. Similarly, variances were observed for the first 4 scales of decomposition, resulting in 48 variances. Finally, the manner in which these parameters are estimated for all scale levels is an expectation maximization algorithm. These values form the final features to be used in classification.

1.2.4 Feature Vector Plots

This feature extraction algorithm was executed on all of the patches used in the testing data set, and the results are included in figs. 3 and 5. Two different examples are shown below, "Brown White Feathers" (figs. 2 and 3) and "Stem" (figs. 4 and 5). These examples were selected due to the quality of the registered patch. In other patches that were sampled, some of the registrations included large areas of black space, not selected for the examples that follow.

The patches from the original piece, from the copy, and from the resulting registered image are displayed side by side. The feature vectors (transition matrix and variances) are plotted with a separate series for the original, the copy, and the registered image. For the Transition matrix, the points plotted in the charts below alternate between values of P_{SS} and P_{LL} . For the variances, the points alternate between large and small variances, σ_L and σ_S , across levels and directions¹.

Each feature (30 transition probability pairs, and 24 variance pairs as outlined above) that is extracted from a single patch counts as an observation in the table. Table 1 displays results for "Brown White Feathers" and "Stem". For these tables, one would expect the registered image to display similar behavior as the copy, since the registered image is an altered form of the copy, just shifted to align with the original during registration. The table displays counts of instances where the registered image had the same relationship to the original as the copy did, and instances where the registered image did not. For example, if the copy had a higher transition probability for P_{SS} than the original, we would group the observations in "Feature Pattern Maintained" group. Otherwise, that count would fall in "Feature Pattern Disrupted". The rest of the feature vectors and associated plots are included in the Appendix.

Figure 2: "Brown White Feathers": Cropped patches to be analyzed



Original

Copy

Registered

¹For ease of interpretation, variances of the feature vectors were log-transformed, so that the scale of the Gaussian variances was comparable to that of the transition probabilities within each patch's feature vector





Figure 4: "Stem": Cropped patches to be analyzed



Original



Сору



Registered





Table 1: Feature Vectors Summarized: "Brown White Feathers" and "Stem"

	Brown White Feathers Feature Vectors Ordered by Magnitude of the Feature Coefficients			Stem Feature Vectors Ordered by Magnitude of the Feature Coefficients		
	Feat. Pattern Maintained	Feat. Pattern Disrupted	Total	Feat. Pattern Maintained	Feat. Pattern Disrupted	Total
$\overline{P_{SS}}$	18	12	30	25	5	30
P_{LL}	21	9	30	24	6	30
σ_L	24	0	24	23	1	24
σ_S	23	1	24	24	0	24

1.3 Results & Conclusions

1.3.1 Data Analysis

The data above exhibit trends within the transition probabilities and large and small variances. The specific results for transition probabilities and variances, as well as the two case studies "Stem" and "Brown White Feathers", are discussed below.

1.3.2 Transition Probabilities

First, we will analyze the transition probabilities. In figs. 3 and 5, the feature vector plots for "Brown and White Feathers" and "Stem", we see that the probability values for both P_{SS} and P_{LL} are typically close to 1. P_{SS} is slightly more variable in "Brown White Feathers" than P_{LL} , while in "Stem" they are both high. Because so many of the transition probabilities are close to 1, our results suggest that the parent is a good predictor of the size of the variance for the child. There is a lower probability of having a parent with a different size variance.

In the transition probability charts, it is rare for the registered value to be smaller than both the original and the copy. The transition probabilities for the originals tend to appear slightly more stable and less extreme than the copies, although this is not always the case.

1.3.3 Variances

When looking at variances, there is a more consistent pattern in the comparison of original to copy and registered. In table 1, summaries of feature vectors of "White Brown Feathers" and "Stem", we see that the feature vector pattern is maintained in all cases except one. In general, the original patches tend to have smaller values for σ_L , when compared to the σ_L in the corresponding copy patches. However, this is not the case for "Stem", for example. The larger variances for σ_L in the "Stem" original patch, compared to the copy, could be due to more variation in the original image, discrepancies in alignment of the patches, or other external factors. Differences such as this one are worth investigating for future analysis. The values for σ_S were more evenly split, with 4 out of the 7 patches having copy with the larger values of σ_S , and 3 with larger values for original.

1.3.4 Conclusions

In general, original image patches tend to have lower values for variance, particularly for σ_L , than do the copy and registered patches. In several of the patches, the values for σ_L and σ_S in the registered patch are both much smaller than even the copy's corresponding values. This makes sense, as the registration process introduces some elements of blur into the image, which would decrease the variance in the high frequency wavelet coefficients. Therefore, the wavelet signatures in these registered images will not be as strong as those in the original or the copy. As such, the registered patches exhibit the same behavior as the copy patches, but with less consistency in the values of both probabilities and variances relative to the original.

2 PART 2

Analyzing the new 2012 data set using Support Vector Machine, with a comparison of different kernel functions.

2.1 Methods

Part 1 of this analysis focused on the different steps in extracting feature vectors from image patches, and on examining the differences between the original, copy, and registered patches of copy onto the original. Because the registered images did not appear to have significantly different features than the original or copy patches, they will not be included in the analysis that follows. In this part of the report, the classification of patches using Support Vector Machines will be analyzed.

2.1.1 Data Sets

For the first part of the analysis, a pair of paintings from the 2008 data set was selected to aid in reproducibility. For the rest of the analysis, paintings from the newer 2012 data set (Caspers, 2012) will be used, since the issues of canvas texture and scanning have been eliminated. This report will discuss results from analysis of Charlotte Caspers's "Nr2_original", and "Nr2_copy", a pair of paintings of a hummingbird on wood board. Additionally, a local artist Jan Dickey painted a copy, "Nr2_Jan", of the original painting. Dickey made the copy using the same materials as the ones used by Caspers in the originals. The three paintings are shown in figure 6. The Hummingbird paintings, after being scanned, are 4049×3894 pixel images.For the pair of Hummingbird paintings, Dickey was given the original and made a copy. For the other pairs of paintings, Caspers did not reveal the ground truth to the team; she did not label the pairs with "original" or "copy". The team lent out the pairs to the other artists, who contacted Caspers separately to identify the originals from the copies. After the copies were finished, the Caspers pair and the finished copy were returned to the team. After analyzing the 3 Hummingbird paintings, where the truth is known, we will additionally analyze a pair of Blue Jays: an original, "Nr1_original", and copy, "Nr1_copy", both painted by Caspers. We will also analyze 3 paintings of a Cardinal: an original and copy painted by Caspers, though the truth is not known for this pair, and a copy painted by Jan Dickey. The two Caspers Cardinals are labeled as "Dot" and "C" for the purpose of this analysis, until the truth is revealed. The Blue Jay scanned images are 6252×4498 pixels, and the Cardinals are 8094×10728 pixels (located in figs. 16 and 17 in the Appendix).

Figure 6: Hummingbird Paintings used in Forgery Analysis



(a) Original - Caspers



(b) Copy - Caspers



(c) Copy - Dickey²

2.1.2 Patch Selection and Pre-processing

Unlike the previous section of this report, for this analysis, an automated process was used to select patches. When creating the copies, both Caspers and Dickey mimicked the style for the foreground, but did not attempt to reproduce the background. For this reason, it was important to avoid selecting patches that contained too much background. To address this,

²Dickey's copy was scanned at a different time than the original and copy by Caspers, resulting in a lower brightness setting. The brightness of that copy was adjusted using Adobe [®] Photoshop [®] software to match the brightness in the original, so that the discrepancies in scanning would not be confounding factors.

a binary foreground/background map was created using Adobe [®] Photoshop [®] software. Patches were generated from random coordinates, and rejected if the corresponding area on the map was greater than 30% background. Furthermore, if more than 5% of a proposed patch overlapped with already-selected patches, it was rejected. The patches were prevented from overlapping to avoid accidental near-duplicate patches in the testing and training sets (i.e. the case where one patch is used in training, and a nearly identical patch is selected for testing). These N patches are divided in half; N/2 patches are assigned as training data, and the remaining N/2 patches are assigned as testing data. After selecting the original N patches from the painting, additional training patches were selected. These additional patches were selected to increase sample size. The $3 \times N/2$ additional patches were selected using the same method as before, only this time the patches were prevented from overlapping by more than 5% with any testing patch. This addition resulted in 2N training patches and the N/2testing patches obtained before. The location of the training and testing patches are shown below in figure 7. The number of patches is configurable, as is patch size. For this analysis, patches were 256×256 pixels³, which allowed N unique patches to be generated using the conditions stipulated above. The number of patches for each painting are displayed in table 2.

 Table 2: Number of Testing and Training Patches For Each Painting

Painting	N (Number of Unique Patches)	Number of Training Patches	Number of Testing Patches	<i>M</i> (Total Patches Analyzed)
Hummingbird	40	80	20	100
Blue Jay	80	160	40	200
Cardinal	120	240	60	300

Figure 7: Location of Testing and Training Patches for Hummingbird. In (d), overlap between testing and training patches is highlighted in yellow.



(a) Unique Patches



(b) Training Patches





(d) Total Patches

As before, all patches were also selected at corresponding coordinates in the copy painting, to create the pair. In the cases where both a Caspers copy and a Dickey copy exist, the same process is repeated for the trio (Caspers original, Caspers copy, Dickey copy), rather than the pair of Caspers paintings.

³Several patch sizes were considered for this analysis. A patch size of 256×256 pixels balanced several considerations: a small enough patch size, such that the number of unique non-overlapping patches was sufficiently large, enough detail in the each patch, such that brush strokes, rather than content, were the primary aspect, and maintaining image quality within each patch without selecting patches that were too small.

2.2 Experiment

2.2.1 Feature Vector Extraction and Transformation

Using the same feature vector extraction methods as in Part 1, a matrix **X** is generated out of the feature vectors for each pair of patches. **X** has $3M \times 108$ coefficient vectors; *M* rows (corresponding to *M* patches) for each of the three paintings, with 108 features for each patch. This matrix has an associated vector **Y** of length 3M, with labels identifying which of the 3 paintings each row of the matrix came from. Pairs or all three of the paintings can be used for comparison, as will be demonstrated throughout the experiment.

As with the first part of the report, the variances in the feature vectors were log-transformed. This introduced some constants in the variance terms that could potentially impact the relationship between values in the feature vectors. To explore this, let W_l represent a wavelet coefficient generated at level l. This coefficient follows some distribution $P(W_l = w_l)$, and has variance $\sigma(W_l)$. When moving to a finer scale, the expectation of the new coefficient W_{l+1} at level l+1 is now equal to the expectation of coefficients at level l, scaled by some constant c, or $\mathbb{E} W_{l+1} = E \frac{W_l}{c}$, and the variance $\mathbb{E} \sigma(W_{l+1})$ can be written as

$$\mathbb{E}\,\sigma(W_{l+1}) = \mathbb{E}\,\sigma(\frac{W_l}{c}) = \frac{1}{c^2}\,\mathbb{E}\,\sigma(W_l) \tag{3}$$

If a log is taken of the variances, then

$$log(\mathbb{E}\,\sigma(W_{l+1})) = log(\frac{1}{c^2}\,\mathbb{E}\,\sigma(W_l)) = log(\mathbb{E}\,\sigma(W_l)) - 2log(c),\tag{4}$$

so that $log(\mathbb{E} \sigma(W_{l+1}))$ is shifted by 2l * log(c) with respect to $log(\mathbb{E} \sigma(W_l))$. To normalize the part of the feature vector corresponding to the variances estimation, these shifts are added back into the log variances of coefficients at each level of decomposition. The logged variances are referred to as a feature vector which is "Un-shifted" in the next sections, and the feature vectors where the variances' constants are added back are referred to as "Shifted".

2.2.2 Support Vector Machines

The feature vectors obtained in the previous step can be used to perform forgery detection, as mentioned in the introduction. The classification of original versus copy will be done using Support Vector Machines (SVM), a type of supervised machine learning used for classification (Winston, 2010). Again, let X and Y be the sets of features and labels, respectively. Denote $(x_1, y_1), \ldots, (x_M, y_M)$ the pairs of features and labels, $y_i \in \{-1, 1\}$. The goal is to find a function of x_i that best maps to the labels y_i , without overfitting (Polatkan et al., 2009). Following Winston (2010) for a description of Support Vector Machines, we set up a classifier, where we classify any feature x_+ , whose label is 1, and x_- , whose label is -1.

$$f(x_{+}) = w \cdot x_{+} + b \ge 1$$

$$f(x_{-}) = w \cdot x_{-} + b \le -1$$
(5)

which can be re-written as

$$y_i(w \cdot x_i + b) \ge 1, \text{ for } i = 1 \dots M \tag{6}$$

In (6), w represents the vector which is normal to the boundary (parallel hyperplanes) of the x_+ features and x_- features. We denote the region containing the x_+ features as the 'positive region' and the region containing the x_- features as the 'negative region'. Additionally in (6), b is some scalar value.

Define x as a support vector if

$$w \cdot x + b = \pm 1 \tag{7}$$

is true. Then let x_1 and x_2 be support vectors such that:

$$w \cdot x_1 + b = 1$$

$$w \cdot x_2 + b = -1$$
(8)

If we subtract the equations in (8), and divide by ||w||, we have an equation for the distance between x_1 , and x_2 , or the distance between two support vectors.

$$w \cdot (x_1 - x_2) = 2 \tag{9}$$

$$\frac{w}{||w||} \cdot (x_1 - x_2) = \frac{2}{||w||} \tag{10}$$

Finally, we want to maximize the distance, found in (10), between the support vectors. This can be done using Lagrange's method, minimizing ||w|| subject to constraints in equation (6).

In the event that the vectors are in high dimensional space, a function, or kernel, is needed to compute the dot products in high dimensional space (Winston, 2010). Support Vector Machines with polynomial and Gaussian kernels were used to classify the unknown patches in MATLAB using the 'fitcsvm' package (MathWorks, 2016b). Specifically, the RBF (Gaussian) kernels, and Polynomials of order 2, 3, 4, and 5 were tested to determine which kernel returns the highest accuracy in results. First, SVM models were created using the training data, and then the model was tested using the testing data. In particular, a test set of patches at the same location of the original painting and the copy are tested independently, so that both of them can be labelled as "Original" or "Copy" simultaneously. The proportion of patches which were correctly predicted as copy or original was recorded for each SVM kernel, and is listed in tables 3 and 4 in the columns labeled 'Regular'. The results are broken down by "Un-shifted" and "Shifted" columns; these show the results for the feature vectors that were log-transformed but not shifted, and for the feature vectors that were log-transformed, and subsequently shifted as shown in eqs. (3) and (4). This process was repeated to incorporate the cross-validation data as well, using the 'predict' package (MathWorks, 2016e), which applies 10-fold cross-validation (MathWorks, 2016f). In this trial, after creating the models using the training data, the cross-validation data was used to ensure the model wasn't overfitting, and finally the model was tested on the testing data. The accuracy proportion using cross validation is shown for the best model in tables 5 and 6 in section $2.2.11.^4$

The classification of patches for test set can be modeled as binomial, where a success is a correctly predicted label for a patch z_i , and k is the number of successes in n trials, or testing patches. The distribution function of Z is shown in (11). Each patch can be considered an independent trial with success probability p. Then, we can model the associated standard

⁴For the CV data, rather than using N/2 testing patches, N/2 training patches, and $3 \times N/2$ extra training, we convert half of the initial training patches to CV. This results in N/2 testing patches as before, N/4 training patches, $3 \times N/2$ extra training patches and N/4 patches for CV purposes. As with the testing patches, the training patches are prevented from overlapping by more than 5% with any CV patch.

error, SE, for the estimate of p using the binomial distribution. The values for the SE are included in addition to the proportions of patches correctly classified in the tables that follow.

$$P(Z = k) = \binom{n}{k} p^{k} (1 - p)^{n - k}$$
(11)

2.2.3 Pairs of "Originals" and "Copies"

The methods above are used to classify unknown patches belonging to an "original" and a "copy". For the purpose of this analysis, we explored several pairs of originals and copies. For the hummingbird, we ran the SVM classification on Caspers's original vs. Caspers's copy, Caspers's original vs. Dickey's copy, and Caspers's copy vs. Dickey's copy. Additionally, one final comparison was made between both Caspers paintings and the Dickey copy. In that trial, all the training patches from Caspers's original and Caspers's copy were included in the training set as originals, and all the patches from the Dickey copy were included as copies. The same method was applied to the testing set for that final comparison. The accuracy results discussed above - proportion of all testing patches which were correctly labeled using the model - for the four different comparisons of hummingbird paintings are summarized in tables 3 and 4.

	Caspers Original/Caspers Copy			Caspers Original/Dickey Copy		
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.5	0.5	0.0790	0.525	0.525	0.0789
2	0.575	0.65	0.0770	0.925	0.925	0.0416
3	0.575	0.6	0.0778	0.9	0.95	0.0416
4	0.55	0.575	0.0784	0.55	0.6	0.0781
5	0.5	0.475	0.0790	0.5	0.575	0.0788

Table 3: Percent of Patches Correctly Predicted - Hummingbird

Table 4: Percent of Patches Correctly Predicted - Hummingbird

	Caspers Copy/Dickey Copy			Caspers Pair/Dickey Copy		
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.575	0.575	0.0781	0.667	0.667	0.0.0745
2	0.95	0.975	0.0300	0.9667	0.9833	0.0246
3	0.95	0.95	0.0344	0.95	0.9833	0.0283
4	0.625	0.75	0.0732	0.55	0.8833	0.0712
5	0.45	0.65	0.0786	0.667	0.5167	0.0777

The same approaches discussed above were applied to the Blue Jay paintings and the Cardinal paintings. Since Dickey did not paint a copy of the Blue Jay, the only comparison is between the two Caspers copies. For the Cardinals, although the ground truth is not known for the pair painted by Caspers, we can still compare them in the same manner as we did for the Hummingbird. In this case, we compare "Dot" and "C", rather than Caspers original and Caspers copy, with a copy of the Cardinal painted by Jan Dickey. The results of these comparisons are listed in the Appendix in tables 13 to 15.

2.2.4 SVM Prediction Scores

In addition to outputting a label for each patch in the testing data, the SVM model in MAT-LAB also outputs a prediction score for each patch (MathWorks, 2016e). The scores represent the distance between the vector for the current patch and the supports of the model. More specifically, if there is a line or region that divides the group into originals and copies, high positive values indicate that the patch was classified far away from the supports of this dividing region, and lower values indicate that the patch was classified in the opposite region. These scores are useful in assessing the degree of certainty that can be attributed to a predicted label. In figure 8, green patches are correctly identified patches, and red patches are incorrectly attributed. It is possible to correctly label the same patch for the original and the copy, to misattribute both patches in that spot, or to correctly predict one, and incorrectly predict the other. The brighter the patch, the higher the prediction score, the dimmer the patch, the closer it fell to the dividing region between the two labels.



(a) Caspers Original

(b) Dickey Copy

Figure 8: Results of patch classification of Caspers original Hummingbird vs. Dickey copy, with polynomial kernel 2 and the final model. Green regions indicate patches which were correctly classified and red patches indicate patches which were incorrectly classified. Brighter patches indicate higher scores in the SVM model.

2.2.5 Feature Selection

A limitation in this initial analysis is the large number of coefficients in each feature vector, when compared to the small sample size of possible patches that can be extracted from the painting (108 vs. M). In the 2008 Princeton analysis, rather than randomly assigning points to the training and testing groups, all patches in certain regions of the painting were assigned to training, and the other half of the painting was assigned to testing (Wang, 2015). In this way, overlap within the training or testing groups was acceptable. If the patches from the top of the painting were all in the training group and many of them overlapped, that would not cause overly similar patches to appear in testing and training. However, this would not result in a completely representative sample of the painting, and the training or testing set could be overly susceptible to unique variations in the painting (i.e. if the eye of the hummingbird appeared only in testing, but nothing of that detailed scale was present in the training set, the model would have difficulty with classification). Instead, reducing the dimension of the feature vector could help resolve this issue. Its effective dimension is smaller than its ambient dimension; therefore, only those features which are important in classification would be retained. For example, perhaps only the fine scale wavelet coefficients would show differences, or variance would be more important than transition probabilities.

2.2.6 Removal of Finest and Coarsest Scales

One method of reducing dimensionality would be to have fewer scales of decomposition included in the classification. Two reductions - one removing the finest scale variances and transition probabilities, and one removing the coarsest scales - were implemented and compared to the results of the original SVM classification. Specifically, transition probabilities between 5 rather than 6 states of wavelet decomposition were observed - yielding 24 total pairs of transition probabilities. Variances were observed for 3 scales of decomposition rather than 4 - resulting in 36 variances, or 84 rather than 108 total features. For both the Hummingbird and the Cardinal, accuracy decreased when removing the finest scale features; it appears to be more important to include those fine scales than to remove them. The results are listed in tables 16 and 17 in the Appendix. We hypothesize that differences between the brush stroke patterns of two artists will become more pronounced when descending from coarsest to finest scales.

Removal of the coarsest scales does not have a negative effect, but rather returns a consistent, or even improved accuracy rate (tables 18 and 19, Appendix). The improvements may result from a lower number of features, without a loss of crucial information which appears to be more present in finer scales. This result was uniform across all 3 paintings. As a result, the coarsest scales were removed from the final model.

2.2.7 Only Variances

A second method of reducing dimensionality is to use only variances, and not include the transition probabilities. Perhaps the scale of the variances is more important than the likelihood of switching from a small to large variance, in which case it would make sense to eliminate the probabilities. Here the results are fairly consistent with those using all features, even though the dimensions had been reduced from 108 features to 48. Some of the results were more variable across kernels than they were for the full features, but this may be because the lower number of features resulted in less overfitting.

2.2.8 Only Probabilities

When retaining only transition probabilities as features rather than variances, accuracy generally decreased across all paintings and combinations. This is to be expected, because the size of the wavelet coefficients should be necessary to distinguish between the original and copy, not just the likelihood of transitioning from a narrow to a wide Gaussian variance. Probabilities seem to be more helpful as an additional measure than as a measure on their own, so it makes sense that they did not work well as the only predictors in the model. This conclusion is further supported by results obtained in section 1.3.2, which showed that most transition probabilities from one hidden state to the same hidden state at a lower scale were high. If it is unlikely to switch to a different hidden state than that of the parent, the transition probabilities will not highlight valuable information in the model. As such, they were removed from the final model. The results from removing probabilities are summarized in tables 20 and 21 in the Appendix.

2.2.9 Lasso Variable Selection

In addition to removing features intuitively, another option to perform feature selection is to use a penalized lasso variable selection in a logistic regression model (Ramsey and Schafer, 2002). As before, if **X** and **Y** are the sets of features and labels, where $y_i \in \{-1, 1\}$, then we could set up a regression model using the features in **X** to predict the labels, **Y**. The form of any generalized linear regression is a response variable, transformed by some function, which we will call the 'link' function, set equal to the explanatory variables multiplied by their respective coefficients, β . Since our response variable is binary, we select a link function accordingly. For a binary response, an appropriate link function is the *logit* function (Ramsey and Schafer, 2002). Let π represent the population probability of the response being 1, then our link function is as follows:

$$g(\pi) = logit(\pi) = log(\frac{\pi}{1-\pi}) = \eta$$
(12)

The regression equation takes the form

$$log(\frac{\pi_i}{1-\pi_i}) = \beta_0 + \sum_{j=1}^F \beta_j x_{ij}, \text{ for } i = 1...M, \, j = 1...F,$$
(13)

which represents the log-odds of the response variable being 1, given the explanatory variables, x_{ij} (Ramsey and Schafer, 2002). For each explanatory variable x_{ij} , *i* represents the patch out of the *M* total patches, *j* represents the feature out of *F* features in the model, β_j is the regression coefficient for each variable, and β_0 is the intercept of the model.

In (13), all features are still included as explanatory variables. As stated above, the goal is to reduce this number of features. Penalized lasso variable selection can be used to retain only those predictors (features), which are significant in the model, given certain constraints. Following Hastie et al. (2009), we want to reduce the number of variables in the model, specifically penalizing models with too many variables compared to those with fewer; this is a type of shrinkage method. Let the objective function be the residual sum of squares from the regression equation. The goal is to minimize the regression residual with respect to the

*l*1 norm of β , a sparsity constraint, as follows:

$$\hat{\beta}^{lasso} = \sum_{i=1}^{M} (\eta_i - \beta_0 - \sum_{j=1}^{F} \beta_j x_{ij})^2$$
subject to
$$\sum_{j=1}^{F} |\beta_j| \le t$$
(14)

where η_i is the response variable from eqs. (12) and (13), β_j and x_{ij} are the same as in (13), and *t* is a chosen constraint. Then the lasso will circle through all combinations of predictors, and retain only those combinations of predictors which can be included in the model without exceeding the constraint in (14) (Hastie et al., 2009). As *t* becomes smaller, some of the β_j values will become 0, and those predictors will not be included in the model. Equation 14 can also be rewritten as follows,

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \{ \frac{1}{2} \sum_{i=1}^{M} (\eta_i - \beta_0 - \sum_{j=1}^{F} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{F} |\beta_j| \}$$
(15)

where the parameter λ is the 'shrinkage parameter', and has a one-to-one correspondence with the parameter *t* from (14) (Hastie et al., 2009).

The R package 'glmnet' was used to perform this variable selection (Friedman et al., 2010; Team, 2015). This package runs the lasso selection, and then selects the value for λ which minimizes $\hat{\beta}^{lasso}$. After setting up the lasso, we employ 10-fold cross validation to adjust the parameters and obtain the final results (Friedman et al., 2010). Any predictors whose β value is 0 can be removed from the model. The selection of variables mostly supported the removal of coarsest scale and finer scale features (only the middle scales were consistently selected), however the variables' inclusion in the model did not follow a decisive pattern. Although the model was able to select certain variables, it failed to converge in most cases, rendering those results null. Often the lasso algorithm is unstable and can be very susceptible to issues with multicollinearity; while ridge or elastic-net variable selection methods may try to shrink the values of the coefficients towards each other for highly correlated variables, the lasso will often pick one of the variables and discard the rest (Hastie et al., 2009). Plots of correlation between feature vectors for a single direction are shown in figure 9. The variables in the plot, variances, alternate between σ_S and σ_L , demonstrating high correlations; etremely high correlation between alternating variances (i.e. between σ_S and σ_S or σ_L and σ_L) indicates that the σ_S at a coarser scale of decomposition is often highly correlated with the same type of variance in a finer scale. Based on the behavior of wavelet decomposition this is expected, but shows that lasso will not work well. Removing just one scale of decomposition was shown to reduce accuracy in section 2.2.6 - to completely remove the multicollinearity would require retaining only one scale of decomposition, which defeats the purpose of the wavelet decomposition.



Multicollinearity for Small and Large Variances

Figure 9: Multicollinearity: Hummingbird Small and Large Variances and 4 Scales of Decomposition. In this plot, *var_i_j_k* corresponds to "variance in the ith direction, at the jth scale of decomposition, where $k = \{a,b\} = \{$ small variance, large variance $\}$.

2.2.10 Fixed and Random Effects Models

A fixed and random effects model could address the multicollinearity issue, by modeling the effect of patch location, direction of the wavelet coefficient, and level of decomposition, as random effects, and the variances or probabilities as fixed effects. In a typical regression model, as in (13), we have only fixed effects, and any additional error associated with the model. When using random effects, we are noting that each value of the random variable should have its own intercept (i.e. each direction of the wavelet coefficients should have its own intercept), rather than allowing all of that variation to simply be attributed to error (Bates et al., 2015). The logistic regression fixed and random effects model is stated below:

$$log(\frac{\pi_i}{1-\pi_i}) = \beta_0 + \beta_j x_{ij} + b_j Z_{ij}, \text{ for } i = 1...M, j = 1...F$$
 (16)

where $\beta_{ij}x_{ij}$ are the fixed effects and b_jZ_{ij} are the random effects, everything else follows from section 2.2.9. To perform this analysis, we used the *lme4*' package in R (Bates et al., 2015). As the two paintings by Caspers are similar, it is not easy to distinguish between

features for the different patches, which results in a long run time to find a model. R typically shuts down on the local machine before the model is able to converge - suggesting that it is not able to separate the two groups and is running indefinitely, or a smaller sample or number of predictors is needed to allow the model to converge. Even after computing the model with just variances, it fails to converge. More investigation should be done regarding these methods, as the main limitations are currently the size of the sample and the processor of the machine; when creating variables for scale, direction, transition probability, variance, and label for each patch in each painting, the data set became very large, which resulted in a memory bottleneck.

2.2.11 Final Model

After investigating the various means of reducing dimensionality, we selected the final model, eliminating the coarsest scale features and retaining only variances. This model has 36 total features. The accuracy comparisons for the Hummingbird paintings, using this final model, are summarized in tables 5 to 8.

	Caspers Original/Caspers Copy		Caspers Original/Dickey Copy			
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.475	0.475	0.0645	0.65	0.65	0.0616
2	0.675	0.725	0.0592	0.975	0.95	0.0245
3	0.6	0.675	0.0621	0.95	0.925	0.0313
4	0.525	0.55	0.0644	0.5	0.775	0.0621
5	0.5	0.575	0.0644	0.5	0.8	0.0616

Table 5: Percent of Patches Correctly Predicted using Final Model - Hummingbird

Table 6: Percent of Patches Correctly Predicted using Final Model - Hummingbird

	Caspers Copy/Dickey Copy			Caspers Pair/Dickey Copy		
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.7	0.7	0.0592	0.667	0.667	0.0608
2	0.975	0.975	0.0202	0.9833	0.9833	0.0165
3	0.95	0.95	0.0281	0.9833	0.9667	0.0202
4	0.5	0.825	0.0610	0.6667	0.86667	0.0546
5	0.5	0.875	0.0598	0.6667	0.65	0.0612

	Caspers Original/Caspers Copy		Caspers Original/Dickey Copy			
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.575	0.575	0.0638	0.575	0.55	0.0640
2	0.525	0.45	0.0645	0.95	0.9	0.0340
3	0.55	0.475	0.0645	0.9	0.825	0.0445
4	0.475	0.475	0.0645	0.525	0.475	0.0645
5	0.5	0.5	0.0645	0.55	0.5	0.0645

 Table 7: Percent of Patches Correctly Predicted using Final Model and CV - Hummingbird

Table 8: Percent of Patches Correctly Predicted using Final Model and CV - Hummingbird

	Caspers Copy/Dickey Copy			Caspers Pair/Dickey Copy		
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.375	0.375	0.0625	0.667	0.667	0.0608
2	0.675	0.85	0.0549	0.8667	0.9333	0.0387
3	0.725	0.85	0.0528	0.8667	0.9167	0.0401
4	0.625	0.825	0.0576	0.6667	0.8167	0.0565
5	0.575	0.6	0.0636	0.667	0.667	0.0608

2.3 Results & Conclusions

In the results above, the accuracy of the SVM kernels vary depending on whether crossvalidation was used, and whether a shift was used. As the sample size was small for some of the paintings, a misclassification of one patch often had a non-negligible impact. For the hummingbird with 20 patch locations used for testing, 40 total patches when comparing pairs of paintings, a change of \pm 0.025 (or 0.017) in accuracy corresponds to one additional or one less patch correctly classified.The values for the other comparisons and paintings are included in 9.

Painting	Comparison	Number of Testing Patches	Accuracy \pm 1 Patch Correctly Classified
Hummingbird	Pair Both Caspers vs. Jan	40 60	$egin{array}{c} \pm \ 0.025 \ \pm \ 0.017 \end{array}$
Blue Jay	Pair Both Caspers vs. Jan	80 120	$egin{array}{c} \pm \ 0.013 \ \pm \ 0.008 \end{array}$
Cardinal	Pair Both Caspers vs. Jan	120 180	$egin{array}{c} \pm \ 0.008 \ \pm \ 0.006 \end{array}$

 Table 9: Accuracy proportion associated with each correctly/incorrectly classified test patch for various paintings

2.3.1 Caspers Original vs. Caspers Copy

When comparing Caspers's original Hummingbird to her copy of the Hummingbird, about 50-60% of patches were correctly labeled. After accounting for error, typically around 5%, this is not a very convincing result. This indicates that the SVM models were not very successful at predicting the difference between the two images based on the input features extracted from the Hidden Markov Tree model and wavelet decomposition. The polynomial kernels 2 and 3 were marginally better than RBF and polynomial kernels 4 and 5. The comparisons between Caspers's Blue Jays, and between Caspers's Cardinals were consistent with the results obtained for the hummingbird (tables 22 to 24 in the Appendix).

2.3.2 Caspers vs. Dickey Copy

When comparing either of the Caspers Hummingbirds to the copy of the Hummingbird by Jan Dickey, the classification was more accurate (tables 5 and 6). Though the RBF and polynomial kernels 4 and 5 also ranged from 50-70% accuracy, the polynomial kernels of 2 and 3 had accuracy rates of about 95%. Overall, the comparison of the Caspers copy to the Dickey copy had slightly better results than the comparison of the Caspers original compared to the Dickey copy. One would expect Caspers's original to be more similar than her own copy would be to a copy produced by a different artist. This more pronounced difference would yield a better distinction for classification, and give the higher accuracy proportion that was observed in the test. The improved accuracy when comparing separate artists was also observed when comparing either of the Caspers Cardinals to the Dickey Cardinal.

2.3.3 Caspers Orignal + Caspers Copy vs. Dickey Copy

To expand the testing set, and to focus the analysis on differences between artists rather than between two specific paintings, we compared Dickey's copy of the Hummingbird to both Caspers's paintings. Patches from both paintings by Caspers were labeled as original, and the Dickey patches were labeled as copies. It is important to note that this caused an imbalance in both the testing and training sets, which now had 2/3 original and 1/3 copy. The results of this comparison were consistent with those from the other two tests between Caspers paintings and the Dickey copy(6). Kernels 4, 5, and RBF tended to have lower accuracy. Specifically for those kernels, values of 0.667 indicate that the model predicted all

originals. This led to 0% accuracy on the copy patches and 100% on the original patches, which supports our hypothesis that having an imbalanced ratio may bias the model. However, for kernels 2 and 3 we see an accuracy rate above 96%, indicating that only 1-2 patches out of 60 in the testing set were incorrectly labeled. Once again, the same comparison using the Cardinal paintings produced comparable results.

2.3.4 Cross Validation

Addition of the cross-validation patches resulted in a lower number of training patches, given that the number of unique, non-overlapping patches is finite. When using CV, although we retained the N/2 testing patches, N/4 patches were removed from training and added to the CV. In tables 7 and 8 we see that accuracy is somewhat lower than it was for the tests using all of the training patches, but still shows the same general trends: the comparison of the two Caspers pairs shows the lowest accuracy, while the other comparisons hover around 90% accuracy.

2.3.5 Overfitting

In general, comparing any of the Caspers paintings to a corresponding Dickey copy results in high accuracy. This could be caused by overfitting in the model. To determine if the final model was overfitting, we generated synthetic data by switching the labels for half of the training patches; half of the originals were labeled incorrectly as copy, and half of the copies were labeled incorrectly as original. For this test we compared the Caspers original Hummingbird to the Dickey copy. If the model were overfitting, we would see results significantly higher than 0.5 accuracy. However, switching the labels did prevent the model from fitting properly, and we observed values around or below 50% in table 25. After accounting for error, this does not provide any evidence that the model overfits on this synthetic data.

2.3.6 Overlap of Testing and Training Patches

Another factor that could contribute to the success of classifying the testing patches is the overlap between testing and training patches. As stated in the methods above, patches were only allowed to overlap by 5% with any other patch, and all additional training patches generated were only allowed to overlap by 5% with a pre-existing testing patch. A lower threshold for overlap (or even a threshold of 0%) would have increased separation or independence between testing and training patches. However, such a low threshold would not be practical as it resulted in too small a sample size, and took too long to run on a local machine. Since some overlap is present, we hypothesized that testing patches with this overlap might be more correctly predicted, with better SVM scores, than those which did not overlap with training patches. An inspection of figure 10 shows some patches with overlapping regions correctly predicted, and also some without overlap which were correctly predicted. In other comparisons of the Blue Jay or the Cardinal, some patches which were incorrectly classified showed overlap with training data, indicating that overlap with training did not appear to pose an unfair advantage in classification. This occurred in figure 10, in the red patch near the bottom of the bird.

Dickey Copy: Regular Unshifted



Figure 10: Overlapping Regions Between Testing and Training Patches, and associated accuracy for Dickey copy Hummingbird, with polynomial kernel 2 and the final model. As before, green regions indicate patches which were correctly classified and red patches indicate patches which were incorrectly classified. Brighter patches indicate higher scores in the SVM model. Grey patches represent location of training patches, and yellow areas indicate overlap between a training and testing patch.

2.3.7 Proximity of Testing Patches to Training Patches

Although overlap may not appear to have a relationship with accuracy, location of testing and training patches might be a factor. If a testing patch is surrounded entirely by training patches, if follows that one would be more likely to predict that patch correctly. Similarly, if there are many testing patches in one area of the painting, and not many training, one would expect the model to perform with lower accuracy in that region. A visual inspection of figure 11 does not produce overwhelming evidence for this phenomenon. The red patch in the lower center was incorrectly predicted, although it was adjacent to several training patches, while multiple patches on the wing tips and extremities of the painting - not surrounded by training patches - were correctly predicted. Additionally, the brightness or SVM score does not seem to be overtly related to the amount of overlap or proximity of training patches.



Figure 11: Overlapping Regions Between Testing and Training Patches and associated accuracy for Dickey copy Hummingbird, with polynomial kernel 2 and the final model. (Same patches continued from figure 10.) As before, green regions indicate patches which were correctly classified and red patches indicate patches which were incorrectly classified. Grey patches represent location of training patches, and yellow areas indicate overlap between a training and testing patch.

2.3.8 Paint Thickness as a Confounding Factor

Since the wavelet analysis relies on examining the patches at finer and finer scales, the brushstrokes seem to provide the main source of inference when creating the wavelet coefficients. However, these brushstrokes are defined not only by their shape and frequency, but by the paint with which they were applied. A closer inspection of the Dickey copy shows it has more thickly applied paint, especially in the belly region of the Cardinal painting. Patches from the same location on the Caspers original and Dickey Cardinal are shown in figure 12⁵. Perhaps patches are more likely to be classified as Dickey because of the paint thickness, rather than because the shape of the brushstrokes is different between Dickey and Caspers. Although this does not indicate that the analysis doesn't work well - variations in paint thickness may be a sign of differing artists - it is not the expected performance or application of the model.

⁵Patch brightness and detail adjusted to highlight differences

Figure 12: 256×256 pixel Patches from the same location in Caspers's original and Dickey copy of the Cardinal to show differences in paint thickness.



(a) Caspers Original

(b) Dickey Copy

3 PART 3

Using one or multiple paintings from the 2012 data set, and the methods outlined in Part 2, to classify originals and copies, where the testing and training sets contain separate paintings.

3.1 Methods

Part 2 of this analysis assessed the strength and accuracy of using Dual-Tree Wavelet transform, Hidden Markov trees, and Support Vector Machines to predict originals from copies. It explored ways to improve accuracy and reduce dimensionality in the model, and analyzed the possible interpretations of the results. In Part 3, we will use the final model produced in Part 2 to distinguish originals from copies, using separate paintings for testing and training data sets.

3.1.1 Data Sets and Preprocessing

Again in Part 3 we will use the hummingbird, Blue Jay and Cardinal from the previous sections. The patches were selected using the same manner as in section 2.1.2, and all sample sizes from table 2 were retained.

3.2 Experiment

3.2.1 Predicting Original and Copy - Cardinal

The goal of this experiment is to determine, from a pair of Caspers paintings, which is the original and which is the copy. Although in Part 2 we were able to distinguish between the pairs of paintings fairly well, this does not give us insight into which one is the original. To test this, we can use other paintings (originals and copies) by the same artists and attempt to predict the difference using those known pairs. As the ground truth for Caspers's Cardinal pair is unknown, we will use "Dot" and "C" as our testing set. The Blue Jay and Hummingbird will be the training set.

To determine which of the paintings is the original, all patches from the Blue Jay pair and all patches from the Hummingbird, as sampled in 2.1.2 of the analysis, are used as the training set. We then used all the patches from the Cardinal pair as the testing set. For the SVM classification, we used the final model obtained in section 2.2.11 in which the coarsest scale is removed and only the variances are retained. Table 10 below shows the results, broken down by painting⁶. Since ground truth is not known, the table shows results if "Dot" were 1s and "C" patches were 0s, for the purpose of analysis.

Caspers "Dot" /Caspers "C"								
	Regular	Un-Shifted	Regular	Shifted				
Kernel	"Dot"	"C"	"Dot"	"C"				
RBF	0.0467	0.9833	0.0467	0.9833				
2	0.3333	0.5833	0.3567	0.69				
3	0.55	0.3733	0.5033	0.42				
4	0.3667	0.66	0.6367	0.4033				
5	0	1	0.74	0.2967				

Table 10: Percent of Cardinal Patches Predicted, for "Dot" as 1s and "C" as 0s, Broken Down by Painting, using Hummingbird and Blue Jay as Training Set (i.e. 0.0467 of all "Dot" patches were predicted as originals, and 0.9833 of all "C" patches were predicted as copies, using the Regular Un-shifted RBF model)

Because the painting of the Blue Jay is larger than the Hummingbird painting, the resulting scanned images are also different in size (the Hummingbird images are 4049×3894 pixels, while the Blue Jay images are 6252×4498 pixels). Therefore, for a fixed patch size such as 256×256 , there are more possible unique patch locations in the Blue Jay than in the Hummingbird (table 2). Including all patches, as sampled in Part 2, from both paintings means we have more Blue Jay patches than Hummingbird patches in the training set. This could bias the model in favor of characteristics or features shown in the Blue Jay and not the Hummingbird, an effect observed with the RBF kernel in section 2.3.3. To address this, an additional test was performed, with equal numbers of patches from the Blue Jay and the Hummingbird in the training set (all Hummingbird patches, but only 200 of the Blue Jay patches from table 2). The results are listed in the Appendix table 26.

3.2.2 Confounding Factor: Size of Patches and Resolution

Classification using multiple paintings as training sets is more susceptible to varying factors than it was for the classification performed in Part 2. The resolution of the patches is different depending on the size of the painting; all patches are 256×256 pixels, but the Hummingbird is small (4049×3894 pixels), the Blue Jay is medium-sized (6252×4498 pixels) and the Cardinal is largest (8094×10728 pixels). The images in figure 13 show the resolution of patches from all 3 birds, of size 256×256 pixels, taken from similar textures on the birds' torsos. Aside from the different resolution, Caspers may have used a different sized brush to

⁶From the totals in table 2, we have 200 training patches from the hummingbird pair + 400 training patches from the Blue Jay pair, and 600 testing patches for the Cardinal pair.

paint finer details in the smaller painting than in the larger ones, further adding variation to the brush strokes observed in each patch.

Figure 13: 256×256 pixel patches from torsos of birds in Caspers's original Paintings to show detail



(a) Hummingbird

(b) Blue Jay

(c) Cardinal

3.2.3 Predicting Original and Copy - Hummingbird

To investigate this scale issue, we tested this painting-level classification on the Caspers hummingbird pair. In this case, we use the medium sized Blue Jay (where ground truth is known), to predict the smaller hummingbird pair. It may be easier to extrapolate from a large painting down to a smaller one, rather than scaling up to a large painting from two smaller ones, as was the case with the Cardinal. The accuracy from predicting the Hummingbird using the Blue Jay as the training set is shown in tables 11 and 12.

	Caspers Original /Caspers Copy									
Kernel	Regular Un-shifted	Regular Shifted	Average SE							
RBF	0.495	0.495	0.0373							
2	0.565	0.545	0.0370							
3	0.575	0.525	0.0371							
4	0.525	0.53	0.0372							
5	0.52	0.485	0.0373							

 Table 11: Percent of Hummingbird Patches Correctly Predicted, using Blue Jay as Training Set

Caspers Original /Caspers Copy									
	Regular U	Jn-Shifted	Regular Shifted						
Kernel	Original	Сору	Original	Сору					
RBF	0.02	0.97	0.02	0.97					
2	0.38	0.75	0.32	0.77					
3	0.38	0.77	0.23	0.82					
4	0.73	0.32	0.52	0.54					
5	0.96	0.08	0.46	0.51					

Table 12: Percent of Hummingbird Patches Correctly Predicted, Broken Down by Painting, using Blue Jay as Training Set (i.e. 0.02 of all original patches were correctly predicted as originals using the Regular Un-shifted RBF model)

3.3 Results & Conclusions

3.3.1 Predicting Original and Copy for the Cardinal

In Part 2, the results vary depending on which paintings are being compared. All comparisons involving a Caspers painting vs. a Dickey painting have high accuracy, leading to confident estimations for both the copy and the original being tested. This is less true for the comparisons of Caspers's originals to Caspers's copies. Accuracy tends to be closer to 60% than 90% in these comparisons. Because comparing two paintings by the same artist is more challenging, we expect our results of this new test to reflect that. Even if the results using the Hummingbird and Blue Jay as training were as accurate as using training patches from the Cardinal itself, we would still experience a maximum accuracy around 60%; this makes identifying the original and copy even more challenging than it would be for a Caspers painting against a Dickey painting.

We see that the models learned are strongly biased. Many of the kernels predict a majority of patches as either originals or copies. When examining at a plot of "Dot" and "C" to see which label was predicted more frequently, we see both paintings have similar distributions, although "Dot" tends to have more values predicted as original, and "C" more values as copy. When separating results by painting, kernels 2 and 3 tended to have more similar rates for the two paintings, while 4, 5 and RBF had extremely different values. This supports previous conclusions that 2 and 3 are the best kernels, because they were not simply predicting one label for all patches, but rather predicting patches with more consistent (if low) results for both paintings.

When we adjusted the training set to have equal numbers of Hummingbird and Blue Jay patches, we see that our results changed slightly, although the breakdown of "Dot" vs. "C" remained consistent, with the model predicting one label more than the other. This change could be the result of smaller sample size and therefore a less representative training set, or the fact that the Blue Jay is a better predictor of the Cardinal than the Hummingbird. This would support our theory that patches need to come from a similarly sized source to be applicable in prediction.

3.3.2 Predicting Original and Copy for the Hummingbird

When classifying patches in the Hummingbird using the Blue Jay as a training set, we observed similar results to those in section 3.3.1. Table 11 shows consistent accuracy values for all kernels, centering around 55%. Kernels 2 and 3 tended to have more consistent accuracy rates when broken down into the two paintings (table 12), compared to 4, 5, and RBF. Those kernels classified a majority of patches in the testing set as originals or as copies.

3.3.3 Confounding Factor: Patch Location

In some comparisons, it seems that the model is predicting based on characteristics of location in the painting, rather than on brushstrokes independent of location. In the pair of paintings in figure 14, we see that the patches in the belly were correctly predicted in the copy, and incorrectly in the original; the model attributed almost every patch of the belly as a copy. This indicates that the characteristics used to build the model are being skewed by location rather than brush stroke pattern. Although this was not an issue in some other models, especially those in Part 2, it is worth noting as a potential limitation of the classification.



(a) Caspers "Dot"

(b) Caspers "C"

Figure 14: Results of patch classification from 3.2.1. Green regions indicate patches which were correctly classified and red patches indicate patches which were incorrectly classified. Brighter patches indicate higher scores in the SVM model. Poor separation of the two paintings is observed.

3.3.4 Using Patch Level Results to Classify a Painting

In the two experiments above, we classified each patch in a pair of paintings individually. We can use those individual results to assess a painting overall - are there more patches classified as originals or copies, and does this provide enough evidence to classify the entire painting as a copy or original? Simply using the majority for each painting is one approach, though it does not incorporate the confidence associated with each patch-level estimate or the accuracy of the model overall. A more informative way to model the true "identity" of a painting, ad hoc, is to use the predicted labels and the SVM scores associated with each patch. If every painting can be modeled with a bimodal distribution centering at 0 and at 1, then it follows that the higher peak of the distribution represents the more frequently predicted label in the painting. We can use the SVM scores as a measure of error associated with each predicted label, and then create a sampling distribution for each painting. Poorly performing models will have two flat peaks that are fairly even, while highly separated models will have one spike and one flat peak. A plot of this type is shown in figure 15 for the predicted Cardinal labels, using the Hummingbird and Blue Jay as training set. In these plots, because of the low accuracy and issues with prediction in the model, we see that both Caspers "Dot" and Caspers "C" favor the copy. Caspers "C" has a slightly higher and thinner spike at 0 and a flatter curve at 1, showing a small improvement over Caspers "Dot". However, this difference is likely not significant given the issues noted above.



Figure 15: Ad hoc distribution of predicted labels for Caspers "Dot" and Caspers "C" Cardinal pair using SVM model with polynomial kernel 2: value of 0 indicates a copy and 1 indicates an original.

4 Future Work

4.1 Patch Size and Sampling Methods

As discussed throughout the report, patch selection and sample size are crucial to obtaining stable results. In the experiments above, a single randomization was used to select patches. An alternative to this, which would maximize space within the overlap thresholds stated above, is to produce multiple randomizations. This would remove some of the variability associated with a spatially inefficient distribution of patches and resulting low sample size. Additionally, it would be useful to run the randomization and feature extraction multiple times, to obtain estimates for the average predicted accuracy and success of the model. Both of these changes would solve some of the issues faced throughout the analysis with sample size.

4.2 Lasso and Mixed Effects Models in R

As described in section 2.2.9, issues of multicollinearity prevented us from creating a successful logistic regression model and performing subsequent variable selection using lasso. A fixed and random effects model could account for the relationship between variables in the features, but in the current analysis was not computable (section 16). In the future, a larger machine will be necessary to run all of the data at once for the fixed and random effects model. In this way, the model would have time to converge, and a subsequent application of the lasso variable selection, other other forms of variable selection could be considered.

4.3 Additional Results Using New Scans

Recently, additional scans were produced of the paintings used in this analysis. For future work, a repeat of this analysis, using those scans would provide useful information. If the results were consistent with the current findings, we could eliminate scanning conditions as a potential source of bias, and increase our sample size to add confidence to our current findings. This is a good way to increase sample size without making any sacrifices of quality or independence, as highlighted in section 2.1.2.

4.4 Transformations

As mentioned in section 1.2.4, the scale of the transition probabilities (0 to 1) and the scale of the variances (frequently greater than 1 or less than 0), provide a potential challenge for distinguishing features. The normalization method used in this report, log-transform, puts the two types of features on a similar scale, but may obscure other trends in the data. Going forward, other normalization methods should be considered and compared to the accuracy of the tests done using the log-transformation.

4.5 Classification at the Painting Level

4.5.1 Modes of Analysis

In section 3.3.4, an ad hoc distribution method was used for classification of paintings, on a painting rather than patch level. Other methods of classification and analysis, such as an ROC curve, could provide additional insight into the validity and accuracy of the model. For future work, modeling of the accuracy and predictions for each painting should be considered.

4.5.2 Image Resolution

In Part 2 of the analysis, the coarsest scales were not as important in classification as the finer ones. Instead of removing the coarsest scale features, another way to achieve this emphasis on fine scale brush strokes rather than coarse scale content is to adjust the resolution of the image. In future analyses, adjusting the resolution of the Cardinal or Blue Jay to match that of the Hummingbird may help resolve some of the discrepancies described in section 3.2.2.

4.6 Other Classification Methods

Beyond feature selection, other classification methods can be considered. K-means and knearest neighbor were also tested in this experiment before performing feature selection (MathWorks, 2016c,d), but the results were inconclusive - likely due to the issue of dimensionality. For k-means, one would divide the testing points into two or more groups, and if there were significant differences between the copy and original points, hopefully they would be highly concentrated in opposite clusters. For k-nearest neighbors, classification could be done based on any point in the data set (or the k closest points) which fell the shortest Euclidian distance away from the point being tested. In both k-means and k-nearest neighbor, now that we have addressed reducing dimensionality, these approaches could be revisited.

Acknowledgements

I would like to thank my thesis advisors, Dr. Daubechies and Dr. Dunson, for their continued guidance on this project. Thank you to Rachel Yin for meeting with me weekly and acting as a phenomenal mentor to me over the past year and half. Thank you to Dr. Cetinkaya-Rundel for advising me throughout my statistics major, and to Dr. Iversen for consulting on models in this project. To all of my friends who helped with coding, I appreciate your constant willingness to help. To my friends and family who provided moral support and read my drafts, thank you again. And finally thank you to my parents, for endlessly encouraging and supporting me over the last 22 years.

References

- Abry, P., H. Wendt, and S. Jaffard (2013). When van gogh meets mandelbrot: Multifractal classification of painting's texture. *Signal Processing* 93(3), 554–572.
- Arganda-Carreras, I., C. O. S. Sorzano, R. Marabini, J. Carazo, C. Ortiz-de Solorzano, and J. Kybic (2006). Consistent and elastic registration of histological sections using vectorspline regularization. *CVAMIA: Computer Vision Approaches to Medical Image Analysis* 4241, 85–95.
- Bates, D., M. Mächler, B. Bolker, and S. Walker (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software* 67(1), 1–48.
- Caspers, C. (2008). Caspers 2008 dataset. http://www.math.princeton.edu/ipai/ datatsets.html.
- Caspers, C. (2012). Caspers 2012 dataset. http://www.math.duke.edu/~rachel/ resources/charlotte2012.html.
- de Freitas, N. (2012). Cpsc340: Hidden markov models. http://www.cs.ubc.ca/~nando/ 340-2012/lectures/16.pdf.
- Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software 33*(1), 1–22.
- Gonzalez, R. and R. Woods (2002). *Digital Image Processing* (2nd ed.). Boston, MA: Prentice Hall.
- Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.
- MathWorks (2016a). Dual-tree wavelet transforms. http://www.mathworks.com/help/ wavelet/examples/dual-tree-wavelet-transforms.html.
- MathWorks (2016b). fitcsvm. http://www.mathworks.com/help/stats/fitcsvm.html.
- MathWorks (2016c). kmeans clustering. http://www.mathworks.com/help/stats/kmeans. html.
- MathWorks (2016d). knnsearch. http://www.mathworks.com/help/stats/searcher. knnsearch.html.
- MathWorks (2016e). predict. http://www.mathworks.com/help/stats/ compactclassificationsvm.predict.html?refresh=true#outputarg_Score.
- MathWorks(2016f).Supportvectormachinesforbi-naryclassification.http://www.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html#bsr5b42.

MATLAB (2015). R2015a. Computer Software.

- Polatkan, G., S. Jafarpour, A. Brasoveanu, S. Hughes, and I. Daubechies (2009). Detection of forgery in paintings using supervised learning. pp. 2885–2888. In Proceedings of the 16th IEEE International Conference on Image processing, ICIP'09.
- Ramsey, F. and D. Schafer (2002). *The Statistical Sleuth: A Course in Methods of Data Analysis* (2nd ed.). Pacific Grove, CA: Duxbury.
- Sapiro, G. (2014). Image and video processing: From mars to hollywood with a stop at the hospital. https://www.coursera.org/course/images.
- Schindelin, J., I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona (2012). Fiji: an open-source platform for biological-image analysis. *Nature Methods* 9(7), 676–682.
- Team, R. D. C. (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing.
- Thévenaz, P., U. Ruttimann, and M. Unser (1998). A pyramid approach to subpixel registration based on intensity. *IEEE Transactions on Image Processing* 7, 27–41.
- Wang, Y. G. (2015). A review of forgery detection in paintings with new discoveries.
- Winston, P. (2010). 6.034 Artificial Intelligence. http://ocw. mit.edu/courses/electrical-engineering-and-computer-science/ 6-034-artificial-intelligence-fall-2010/.

Appendix

For Feature Vector Plots from section 1.2.4, see attached folder. For all MATLAB and R scripts, see attached folder.



Figure 16: Blue Jay Paintings used in Forgery Analysis

(a) Original - Caspers

(b) Copy - Caspers

Figure 17: Cardinal Paintings used in Forgery Analysis



(a) "Dot" - Caspers



(b) "C" - Caspers



(c) Copy - Dickey

Caspers Original/Caspers Copy								
Kernel	Regular Un-shifted	Regular Shifted	Average SE					
RBF	0.5375	0.5375	0.0557					
2	0.9625	0.9625	0.0212					
3	0.925	0.9	0.0315					
4	0.6125	0.6625	0.0537					
5	0.5875	0.462	0.0558					

 Table 13: Percent of Patches Correctly Predicted - Blue Jay

 Table 14: Percent of Patches Correctly Predicted - Cardinal

	Caspers "Dot" /Caspers "C"			Caspers "Dot"/Dickey Copy		
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.5917	0.5917	0.0366	0.6	0.6	0.0365
2	0.7083	0.675	0.0344	0.975	0.9667	0.0125
3	0.667	0.65	0.0353	0.7833	0.95	0.0253
4	0.466	0.6	0.0372	0.5083	0.825	0.0351
5	0.5	0.4917	0.0373	0.525	0.5083	0.0372

	Caspers "C"/Dickey Copy			Caspers Pair/Dickey Copy		
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.7583	0.7583	0.0319	0.667	0.667	0.0351
2	0.95	0.9417	0.0169	0.9611	0.9667	0.0139
3	0.75	0.933	0.0272	0.9667	0.9444	0.0154
4	0.4917	0.8417	0.0351	0.5611	0.7056	0.0359
5	0.5167	0.5	0.0373	0.3778	0.6333	0.0373

 Table 15: Percent of Patches Correctly Predicted - Cardinal

 Table 16: Percent of Patches Correctly Predicted without the finest scale - Hummingbird

	Caspers (Original/C	aspers Copy	Caspers Original/Dickey Copy			
Kernel	Regular Un-shifted	Regular Shifted	Regular Average Shifted SE		Regular Shifted	Average SE	
RBF	0.475	0.475	0.0789	0.55	0.55	0.078660664	
2	0.5	0.55	0.078958058	0.725	0.7	0.071561818	
3	0.475	0.6	0.07883428	0.8	0.775	0.064680706	
4	0.5	0.55	0.078958058	0.625	0.675	0.075415516	
5	0.475	0.5	0.079032232	0.55	0.675	0.077029824	

	Caspers Copy/Dickey Copy			Caspers Pair/Dickey Copy		
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.65	0.65	0.0754	0.667	0.667	0.0745
2	0.85	0.875	0.0545	0.85	0.833	0.0577
3	0.85	0.875	0.0545	0.85	0.8667	0.0551
4	0.75	0.825	0.0647	0.7333	0.8	0.0669
5	0.65	0.7	0.0741	0.5667	0.7333	0.0754

 Table 17: Percent of Patches Correctly Predicted without the finest scale - Hummingbird

 Table 18: Percent of Patches Correctly Predicted without the coarsest scale

 Hummingbird

	Caspers Original/Caspers Copy			Caspers Original/Dickey Copy		
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.425	0.425	0.0782	0.7	0.7	0.0725
2	0.65	0.7	0.0741	1	1	0.0000
3	0.675	0.65	0.0748	0.975	0.975	0.0247
4	0.5	0.675	0.0778	0.5	0.65	0.0782
5	0.5	0.5	0.0791	0.5	0.5	0.0791

	Caspers Copy/Dickey Copy			Caspers Pair/Dickey Copy		
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.7	0.7	0.0725	0.667	0.667	0.0745
2	0.975	0.975	0.0247	0.9833	1	0.0144
3	0.95	0.975	0.0300	0.9833	0.9833	0.0203
4	0.5	0.75	0.0765	0.6667	0.8833	0.0660
5	0.5	0.55	0.0790	0.667	0.667	0.0745

 Table 19: Percent of Patches Correctly Predicted without the coarsest scale

 Hummingbird

Table 20: Percent of Patches Correctly Predicted using only probabilities - Hummingbird

	Caspers Original/Caspers Copy			Caspers Original/Dickey Copy		
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.45	0.45	0.0787	0.6	0.6	0.0775
2	0.55	0.55	0.0787	0.625	0.625	0.0765
3	0.575	0.575	0.0782	0.625	0.625	0.0765
4	0.575	0.575	0.0782	0.625	0.625	0.0765
5	0.575	0.575	0.0782	0.625	0.625	0.0765

	Caspers Copy/Dickey Copy			Caspers Pair/Dickey Copy		
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.575	0.575	0.0782	0.667	0.667	0.0745
2	0.625	0.625	0.0765	0.6	0.6	0.0775
3	0.625	0.625	0.0765	0.65	0.65	0.0754
4	0.65	0.65	0.0754	0.65	0.65	0.0754
5	0.65	0.65	0.0754	0.6167	0.6167	0.0769

Table 21: Percent of Patches Correctly Predicted using only probabilities- Hummingbird

Table 22: Percent of Patches Correctly Predicted using Final Model - Blue Jay

	Caspers Original/Caspers Copy								
Kernel	Regular Un-shifted	Regular Shifted	Average SE						
RBF	0.625	0.625	0.0541						
2	0.9625	0.9625	0.0212						
3	0.9375	0.925	0.0283						
4	0.775	0.5375	0.0531						
5	0.5625	0.85	0.0509						

	Caspers "Dot" /Caspers "C"			Caspers "Dot"/Dickey Copy		
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.5583	0.5583	0.0370	0.7583	0.7583	0.0319
2	0.6333	0.6917	0.0352	0.9667	0.9583	0.0142
3	0.6417	0.575	0.0364	0.9667	0.9583	0.0142
4	0.5333	0.4833	0.0373	0.8	0.75	0.0311
5	0.4833	0.5417	0.0373	0.5	0.8167	0.0353

Table 23: Percent of Patches Correctly Predicted using Final Model - Cardinal

 Table 24: Percent of Patches Correctly Predicted using Final Model - Cardinal

	Caspers "C"/Dickey Copy			Caspers I	Pair/Dicke	у Сору
Kernel	Regular Un-shifted	Regular Shifted	Average SE	Regular Un-shifted	Regular Shifted	Average SE
RBF	0.7833	0.7833	0.0307	0.6833	0.6833	0.0347
2	0.9833	0.975	0.0106	0.9667	0.9722	0.0128
3	0.9583	0.9583	0.0149	0.9167	0.95	0.0186
4	0.8167	0.6833	0.0323	0.6778	0.8944	0.0306
5	0.5	0.75	0.0361	0.6667	0.7389	0.0341

Caspers Original/Dickey Copy				
Kernel	Regular Un-shifted	Regular Shifted	Average SE	
RBF	0.35	0.35	0.0616	
2	0.475	0.5	0.0645	
3	0.375	0.325	0.0616	
4	0.525	0.5	0.0645	
5	0.5	0.35	0.0638	

Table 25: Percent of Patches Correctly Predicted using the final model and synthetic data as described in section 2.3.5 for Hummingbird. In this test, half of the training labels were reversed.

Table 26: Percent of Cardinal Patches Predicted, for "Dot" as 1s and "C" as 0s, with equal patches from the Hummingbird and Blue Jay in the Training Set

Caspers "Dot" /Caspers "C"				
Kernel	Regular Un-shifted	Regular Shifted	Average SE	
RBF	0.5233	0.5233	0.0204	
2	0.4667	0.5067	0.0204	
3	0.4533	0.4467	0.0203	
4	0.5767	0.5167	0.0203	
5	0.5	0.53	0.0204	