“Morelli Machine”: A Proposal Testing a Critical, Algorithmic Approach to Art History

**CO-PIs:** Christopher J. Nygren and Alison Langmead

**PI Institution:** University of Pittsburgh

**ECSS Team:** Alan Craig, Paul Rodriguez
Morelli Machine

Abstract
The Morelli Machine refers to an algorithmic approach to characterizing authorship from the late 19th century which proposed that fine details of minor items in a painting would reveal particular styles. The PIs set out to test the hypothesis that contemporary computer vision techniques could perform this sort of “stylistic” matching. In order to do this, they sought to mechanize a method that is indigenous to art history and that uses details as a proxy for style. This project approached the question of “style” as one of extracting features that have some discriminatory power for distinguishing paintings or groups of paintings. We used feature discovery from a pretrained convolution network (VGG19) for object recognition. We processed both whole images and some class of image parts (ie mouths), and performed clustering. In this presentation I will review the image preparation steps, extraction steps, clustering results, and cluster evaluation. The upshot is that all convolution layers indeed have discriminatory features, and different layers might have different kinds of features, with different interpretability that may be hard to define.
Motivations

• Study algorithmic approach to ‘stylistic’ elements

• Perform analysis for both whole images ‘holistically’ and for certain kinds of details, ie face parts, for detailed analysis

• Evaluation the applicability of computer vision techniques, including Convolution Networks, to issues in art history
Goal

• Discriminate paintings and face parts of humans in those paintings with the kind of features useful in computer vision
• Analyze the features
• Show the grouping based on those features in a set of historical paintings
Intro

• Computer vision techniques for object discrimination rely on either predetermined features or feature discovery

• Predetermined features:
  • Edge detector (ie pixel patterns that indicate an edge)
  • Histogram of gradients (ie count up pixel intensity changes in different directions around each point)
  • Entropy (ie statistical measure across all pixels)

• Feature discovery: Convolution Neural Networks learns filters that act as feature detectors
  • Lower layers reflect texture, color, and intensity variations
  • High layers reflect object like features
  • Pretrained networks with 1000 objects are readily available
VGG19 – pretrained CNN for object recognition

Extract max-pool convolution layer maps as features
Layer 1: 64 maps of 112x112
VGG19 – pretrained CNN for object recognition

Extract max-pool convolution layer maps as features
Layer 1: 64 maps of 112x112
2: 128 56x56
3: 256 28x28
4: 512 14x14
5: 512 7x7
Which convolution layer to use?
Research using CNN for mask output shows that lower layers are important to for finer details – so use all convolution layers.
Image samples

- Images are high resolution (1K-4K height and widths) but some have frames, and frames are differently shaped
Image preprocessing for pretrained VGG19

- VGG19 uses 224x224 images
- So images need to be resized and frames cropped out (otherwise frames are a dominating feature for the computer)
- 4 resizings (shortest-side pixel length: 224 to 450)
  - X
  - 3 cropping (center, +/- off center)
Preprocessing Steps:
1 Take 12 different combination of resize and crop

2 Select the best one that centers the subject, leaves out frame, without chopping off faces

This is much quicker than trying to manual select a cropping/resize/frame extraction

A few (~20) images were deemed too hard to get a square cropping (e.g. triangular frames)
Preprocessing features

• 1821 images were run through VGG19 network

• Dimension reduction: Each map in each layer was summarized by taking the norm (ie sum square) and scaled by log – this leaves 1 number for each map and 1821 x 1472 data matrix
  • Helps reduce dimensions for each layer (ie 512 maps of 7x7 => 512 values
  • Avoids mixing together all maps before a decomposition so that each feature map (and each underlying filter) is represented
  • Reflects the intuition that each feature map is a detector for the presence of some feature (eg. Chollet)
Cluster procedure

• K-means cluster was applied to the data matrix

• K (number of clusters) selection
  • K was evaluated using Silhouette test, and Gap test (matlab evaluation function)

  Silhouette test typically showed 3-4 clusters
  Gap test showed typically showed 20-40 clusters

Based on tests with simulated data, this indicates that after 3 or 4, there are many possible clusters, though not necessarily well separated

A hierarchical clustering was used to help select a reasonable K
Cluster procedure

- Hierarchical clustering was applied to compare increasing K for 4, 8, 16, 32, and 16 was selected as a reasonable groupings (with PI consultation)

A dendrogram showing top 32 leaves in a hierarchical clustering
Eg: 40 images closest to cluster 1,2 centers (along with original image thumbnails)

Frame elements still there perhaps, but not dominating
Also, some duplicates were left in the data
Do summary features capture data OK?

- The summary features for individual layers form $N \times P$ submatrices, for $N=1821$, $P$ in \{64, 128, 256, 512, 512\}.

- The total feature maps for individual layers form $N \times P \times H \times W$ matrices, where $H \times W$ is height and width of each feature map in layer.

- The 1\textsuperscript{st} principle component of the summary matrix is highly correlated with (.6-.9), but with lower variance than, the 1\textsuperscript{st} component of the full matrix.

**The summary matrix does maintain a lot of information in a feature map (and reduces dimensions)**
Evaluating features

• Does a feature help discriminate between any pair of clusters?
  • Run 1472 x 16 x 16 2 sample ttests

• Almost all features have a small (corrected) p-value for at least 1 pair

• **Implies that all features have discriminatory information**

• Also makes sense given the varied clusters
Evaluating features

• Which features are most discriminatory for each cluster?
  
  • For each pair of clusters sort 1472 pvalues.

  • The most discriminating features come from layers 3,4,5

  • Implies that clustering is based on those features combinations relevant for object recognition

  • Conversely, the images have salient objects (or object like items)
Classification Labels have much overlap

Cluster 1, members 85, top 3 labels (average score)
0.13 "vestment"
0.09 "cloak"
0.08 "book jacket, dust cover, dust jacket, dust wrapper"

Cluster 2, members 227, top 3 labels (average score)
0.18 "vestment"
0.06 "altar"
0.06 "throne"
Comparing layers to predetermined features

• Extracted several measures that summarize an image
  • Histogram of Gradients (HOG) - over several spatial scales, sum of number of pixels with ‘strong’ gradients, maximum among scales
  • Entropy
  • Edge detector (Sobel and Canny) – sum of number of pixels at an edge
  • These also lose spatial information (as do CNN summaries)
Comparing layers to predetermined features

• Extracted several measures that summarize an image
  • Histogram of Gradients (HOG) - over several spatial scales, sum of number of pixels with ‘strong’ gradients, maximum among scales
  • Entropy
  • Edge detector (Sobel and Canny) – sum of number of pixels at an edge
  • These also lose spatial information (as do CNN summaries)

• HOG is most correlated with higher CNN layers (eg 0.36 with layer 3)
• Edges and Entropy are most correlated with lowers layers (~0.45-0.50 with layer 1)

• Within layers, abs. correlations avg: 0.58 (layer 2) to 0.12 (layer 5)
• Between layers, abs. correlations avg: ~0.54 (nearby layers) to 0.15 (w/layer 5)
Clustering with layer 1 alone
Clustering with layer 1 alone – textures more relevant
Other vis – tSNE plot and thumbnails

Colored by cluster, corresponding thumbnails,
Other vis –tSNE plot and thumbnails

Colored by cluster, corresponding thumbnails, but hard to show too many, hard without interactive zoom and pan
What if every feature is a ‘stylistic’ element? 
try: normalize all features by mean and standard deviation, 
take feature with highest outlier, gather images that are outliers in that feature

Example: Feature 877, from layer 4, is most active for the following images 
(top 5, followed by next 10)

Is some bright yellowish-white spot a stylistic feature?
• Also produced spreadsheet with summary information, including inter vs intra distances for each image (ie ‘Silhouette’ measure), class id, file name, etc.
• PIs have spreadsheet with meta data for other possible ways to combine and organize data.
Processing Face Parts

• Google Vision Api was run to extract faces (3205 faces)
  • Performed better than well known DLIB library, OpenCV
  • Required resize larger images b/c of memory limitations
  • Returns landmarks, which were post-processed to extract eyes, nose, mouth
  • Pose (ie head turn) estimates used to adjust extraction boxes
  • Performs best with straight-on face, less well with profiles
  • Mouths extracted easiest, used in clustering
Mouth clustering

• Mouth had low object salience
• Many images needed to be up-sized. Mirror padding might work better.
Hand Extraction

• We reviewed several methods for hand detection and segmentation
• An older work using edge detectors and (part-based) segmentation performed poorly, but did provide some sample training sets with bounding boxes (Mittal, etal 2011)
• Methods:
  faster RCNN for region identification
  person detectors and face detectors to get hand-related information
  Facebook’s Detectron for human body key-point detection
CNN to classify and detect: Faster RCNN
(Ren, He, Girshick, Sun)

A side path to select Regions
CNN to classify and detect: Faster RCNN

(Ren, He, Girschick, Sun)

1. Start with 2000 sampled regions; segment, group, get texture of possible foreground regions

2. Then use feature map values and a model to predict if an object (of any class) would be detected in each window.

3. Pass maps for best regions to classifier

Pool with adjustable windows to transform into one size vector for classification

Images are normalized and resized
Region output

- Output bounding box information (box center, height, width)
Samples of ‘hand’ training data

Person boxes (using YOLO in ‘darknet’ code)

Face detection using ‘DLIB’ library

Hand boxes were hand drawn (Mittal et al.)
RCNN results (training from scratch)

- Using 50% overlap with true box as correct ~50% TP rate
- ~4-6 hours on 1 gpu node (6 tasks) 5-10 epochs on ~1K images (proposing boxes can take a long time)
Caffe2, Facebook “Detectron” networks

Object Detection
ie getting a region bounding box (rcnn)

Object Segmentation
ie getting a mask (mask-rCNN)

Object Parts
ie getting keypoints (keypoint-rCNN)
Based on person box, and keypoints for elbow wrist, a guess of hand position can be made. The exact position could be improved with pixel intensity levels related to the face, but that might introduce a bias. Also, potential issues include: other faces, bent hands, occluded hands, hands holding objects, etc...
Other explorations:

• Using cluster results, ran decision tree to classify images
  With some pruning you can identify the ‘hard-to-classify’ images – eg
• Combining with features by cluster -
  19 of 41 features that best discriminating features are also in top 50 important variables for random forest algorithm

• Future work could run ‘heatmap’ type analysis to see what part of an image activates those features the most (but maybe not the ‘dreamscape’ ones)
Other explorations:

• Network analysis based on similarity (or correlation), to gather centrality measures. But distance is not a natural link.
• Continue hand analysis
• Apply clustering to eyes and nose
• Use Suave (suave.sdsc.edu  I. Zaslavsky) for sorting through thumbnails (in process)
Summary

• Computationally: training benefitted from GPU nodes, but otherwise, for extracting layer activations and clustering, jobs required ~1-2 hour with single CPU nodes

• How do features relate to ‘style’ is still subjective, but at least we can point to computer visual elements that make distinctions

• CNNs open up a lot of possible questions and avenues for different domains