CS 591 E2: Optimization Methods and their Applications – Fall 2016

Image Categorization via Winnow Algorithm with Weak Classifier Inputs

Mike Mangione

1 Abstract:

Machine Learning and Computer Vision are two fields that are developing at a rapid pace and stand to improve the other when working in conjunction. One of the more consistently studied Computer Vision tasks is the identification of remarkable or otherwise significant portions of an image, and is often described as Image Saliency. Image saliency algorithms are the backbone of many more complex computer vision tasks, ranging from image categorization to pattern matching. Saliency computation, like many computing tasks, contains an implicit tradeoff: accuracy for runtime. Many saliency models that possess a quick runtime are usually inaccurate due to their general tendency to focus on a finite number of image features. However, I expect the collective estimation of models using these low-feature models would be significantly more accurate than the individual models. For image classification, the use of an optimization method for a set of binary inputs can quantify the likelihood of a saliency model to correctly identify a category of image. In this paper, I detail the development of a program that uses a modified implementation of the Winnow algorithm to improve the collective prediction of three common saliency computation algorithms. The resulting Winnow outputs indicate that certain saliencies are better tailored to specific tasks, and their combined efforts can produce a measurably improved categorization of the test image.

2 Introduction:

Over the course of the semester, we have covered a variety of algorithms designed to enhance the speed and accuracy of automated predictions across a number of problem spaces. I was previously interested in computer vision, a field of computer science that directly uses many machine learning principles. I had previously been exposed to some image categorization and dictionary learning problems, which involves the sorting of testing images into predefined categories trained (typically) on manually labeled image sets. At the core of the image categorization problems is the need to determine what parts of an image can determine their category, and how best to manipulate the testing image to expose these properties. This need is fulfilled using image saliency models, series of algorithms and modifications designed to capture the important parts of an image. Having read about a number of saliency models in previous courses and being exposed to Winnow, a boosting algorithm which takes arrays of binary elements as input, I decided to implement a Winnow-based analysis of saliency models and their effectiveness in classifying images. Additionally, I used this Winnow analysis in realtime to make a multi-saliency category prediction, and evaluated that prediction with the Winnow2 algorithm.

3 Saliency Models:

In order to better understand the variation between saliency models, it is worth briefly discussing their motivations and implementations.
The Hou saliency model is a matrix decomposition using the log frequency curve of a sample image. The process starts by running a two-dimensional fast Fourier transform, essentially producing two vectors corresponding to the frequency distribution of the image by its height and its width. The magnitudes and angles (or in terms of image processing, intensities and frequencies) are computed from the Fourier transform. From here, the log of the magnitude is recorded, and the output is the log spectrum curve of the image. From here, the matrix decomposition works as follows:

Algorithm 1 Spectral Residual Computation

```
Parameters: $A \in \mathbb{R}^{m,n}$ where m = maximum log intensity of image, n = maximum frequency of image
  $T \in \mathbb{R}^{m,n}$ is zero matrix
  for $i \in \mathbb{R}^{m,n}$ do
      $T_i = (\sum_{t=1}^{3} \sum_{u=1}^{3} A_{tu})/9$
  end for
  return $A - T$
```

Hou is a bottom-up saliency, which means it extracts features from an unprocessed image. In this case, the fast Fourier transform frequency of the image is itself the feature. Bottom-up models are naturally in opposition to top-down models, which use a priori weights and modifications to the base image to affect the saliency, explicitly through additive and multiplicative boosting of desired features. One way to think of the difference between the two is bottom-up saliency looks for outliers in the basic image and builds a saliency from these outliers, whereas top-down saliency defines components of the image to search and from the results it decides what should be considered an outlier.
The Itti model is a such top-down saliency, in that it uses color, intensity, and image orientation as factors for computation. While the Hou saliency uses one factor to pursue salient object detection in its top down approach, Itti uses three preset factors and a series of convolutions that produce a final saliency. These convolutions can be broken into four main parts:

(i) Feature maps, the initial computations of salient features for color, intensity, and orientation

(ii) Conspicuity maps, computations run on the feature maps to select the most prominent features

(iii) Saliency maps, which is the aggregate feature collection the conspicuity maps

(iv) Inhibition of return, or the normalization of the output

The benefit of the Itti model is a saliency that is not strictly defined by a single aspect of the image, and with a number of layers of convolution you are more likely to get an output that is representative of a number of pertinent components of an image. As such, Itti works very well on images that have few obvious focal points, especially if these points have a few qualities that make them stand out (e.g high/low intensity values, significantly different coloring than the background, etc). This sounds well and good, but these same considerations make the Itti

model a poor identifier of large salient objects (e.g. an image of a beach, or a forrest).
The final model for consideration is called Graph-Based Manifold Ranking, and it is a kind of bottom-up saliency computation. A brief outline is included below: One unique characteristic of

Algorithm 2 Bottom-up Saliency based on Manifold Ranking

```
Input: an image
1. Segment the image using superpixels, construct a graph using the superpixels as nodes, and compute the degree matrix D and the weights matrix W
2. Compute $(D - W)^-1$ and set the diagonal to zero.
3. Sample one node on each image side, create four different patches of super pixels representing the sides of the image, and compute an image saliency map using the features four sides as saliency criterion. This produces a saliency map of the "background" of the image.
4. Bi-segment the background saliency to extract potential foreground nodes, and compute their saliency using foreground nodes $F$:
        $(Di - Wi)\forall i \mid i \in F$
```

manifold saliency is that it explicitly attempts to consider the background and the foreground of the images as two distinct entities. This should work fine for images that capture the entirety of the subject in the frame, but for larger subjects this becomes troublesome. For example, ocean pictures will almost certainly fail to capture the entire body of water in the frame, and the characteristics of the water might be considered part of the background. Many of the resulting saliencies are indistinguishable for this very reason.
That said, manifold saliency works well on moderately sized subjects entirely encapsulated in the frame, including the car images used in this implementation. The superpixels approximate the features of a car well, due to its preexisting geometrical features.

3.1 Superpixels:

Superpixels themselves are a means of creating a generic representation of an image. By evaluating the differences between sequential pixels, superpixel algorithms can create grouping of physically close pixels that share similar color values or intensities.
During the development of the Winnow experimentation code, superpixels were at one point overlaid on the computed saliencies to produce a further generalized image. This made the saliency models even more inaccurate, and was subsequently removed from the process.

4 Winnow1 Algorithm:

As discussed in class (and enumerated in other resources), individual inputs to the winnow algorithm are boolean valued, so we can say the instance space is $X = 0, 1^n$. For each of the instances, we initially assign a weight of 1, so we can create an array the length of the number of instances, initializing each to 1 when the instance is recorded.
After the first instance is recorded, an oracle predicts the potential of the model to achieve a correct result. This prediction is typically a summation of the product of the instance value by a preset parameter α.The Winnow1 algorithm is described below:

Algorithm 3 Winnow1 Algorithm

Parameters: $G \in \mathbb{R}^T$ where T = the number of (chronological) predictions made by the expert, and $\forall i \in T, G_i \subset [0,1]$

```
$P \in [0,1]^T, P_1 = 0$
$\alpha = 2$ if $P_i = 0, \alpha = 0$ if $P_i = 1$
for $i \in \mathbb{R}^T$ do
    if if $G_i \neq P_i$ : then
        for $\forall j \mid j \leq i$ do
            $\theta = j/2$
            $G_j = \alpha G_j$
        end for
    end if
    $P_{i+1} = 1$ if $\sum_{t=1}^{T}(G_t) \geq j/2$
    $P_{i+1} = 0$ otherwise
end for
```

4.1 Itti and Winnow:

Of the three Saliency models, Itti has a number of qualities that make it uniquely suited to optimization by the Winnow algorithm. Namely, Itti's use of color, intensity, and object orientation can be considered individual experts, and as separate channels they can execute the Winnow algorithm. The output of such a configuration would be the likelihood of a certain characteristic of the image being able to correctly classify the input.

One way of testing this framework would be to select one of the multi-channel stages (feature, conspicuity, or saliency mapping), isolate the inputs and run the convolution to completion using the isolated channel. From there, the classification result would be run against the Winnow algorithm, and the output might be used in the next Itti iteration to boost high classification accuracy features.

Another significant quality of the feature mapping stage is that the output is 3-literal monotonically disjoint. This means that the output can be separated into the three features perfectly, and that a feature's contribution to the output is not the result of a negation. A counter example might be a saliency model that takes the features of color and orientation, but uses an inverse feature map of the intensity.

As such, we can utilize Littlestone's analysis of the mistake bounds of Winnow1 to conceive how many mistakes can possibly be made at each iteration, a way of understanding its convergence. For our 3-literal monotone disjunction Itti feature map, Littlestone's mistake bound is as follows:

$$3\alpha(\log_\alpha(\theta) + 1) + n/\theta$$

For Winnow1 specifically, we assume $\alpha = 2$ and $\theta = n/2$:

$$6(\log_2(n/2) + 1) + 2$$

$$6\log_2(n) + 2$$

Logarithmic mistake bounds make for resilient predictions, the use of a constrained α reduces the bound even further.

5 Experimentation:

In order to evaluate the efficacy of both the saliency models and the winnow algorithm, I implemented a simplified dictionary learning framework. The overarching structure of the basic framework is as follows:

(i) Collect training images and compute the saliency of each, for each model

(ii) Scale training saliencies such that saliency $S \in \mathbb{R}^{100*100}$

(iii) Sequentially compute the saliency of each testing image, for each model

(iv) Scale test image T such that T $\in \mathbb{R}^{100*100}$

(v) For each category $C \in [S]^n$, compute $argmin(S_i \in C)\|S_i - T\|_1$

(vi) Categorize T in the category with the minimum L1 difference

(vii) Run Winnow against the result for each category and each model

Deciding between the L1 and L2 norm came down to which was more robust. In theory, the L2 norm should perform slightly better given nuanced training data and categories, but after running a few test images it became apparent that the training data was not adequately tailored.

5.1 Input Data:

Two fundamentally similar sets of data were collected for this project, namely the three distinct sets of training images and a single set comprised of testing images. A total of 164 waves images, 79 car images, and 104 cat images comprised the training data sets. At first, two additional categories (gears and the letter E) were included in both data sets. However, image quality and consistency of output became an issue, and they were redacted for the main experimental data collection.
These categories were considered because of their distinct image traits. Pictures of the ocean are generally at distance, comprised of one color, and generally have homogenous intensity values. Pictures of cats are usually up close, vary in orientation, and have distinct color differences. Pictures of cars are usually at medium range, have stark color differences, and usually are in the same general shape.
Images were screened for content and estimated categorization accuracy, but future tests would benefit from only selecting images that have a high correct classification probability from more advanced image classification algorithms and neural networks. Google's Multibox Object Detector is one such deep neural network for object detection, though its composition is admittedly beyond the scope of this project.

5.2 Revising Winnow:

Shortly after implementing the Winnow algorithm, it became clear that the saliency model predictions were much less accurate than anticipated. Some of the error can be attributed to the lack of pruning for training images, as well as the addition of complex testing images.
This low success rate for the saliency models proved undesirable for the low-tolerance Winnow1 framework. Resetting the array of predictions on a false-positive Winnow result caused all the

models to converge on zero after less than 20 iterations. For saliency models that we expect to be inaccurate, we want to maintain some record of past results, albeit scaled to reflect the accuracy of the algorithm.
As such, the Winnow2 algorithm described below was implemented:

Algorithm 4 Winnow2 Algorithm

Parameters: $G \in \mathbb{R}^T$ where T = the number of (chronological) predictions made by the expert, and $\forall i \in T, G_i \subset [0, 1]$

```
  $P \in [0,1]^T, P_1 = 0$
  $\alpha = 2$
  $\alpha \leftarrow [\alpha \text{ if } P_i = 0, 1/\alpha \text{ if } P_i = 1]$
  {execute Winnow1}
```

Winnow2 provides a suitable framework for algorithms with high variability in accuracy. However, since α remains constant for both false successes and false failure predictions, Winnow2 "converges" on two alternating values for low accuracy functions. The ideal Winnow variation for evaluating saliency models must distinguish algorithms with low accuracy from algorithms with no chance of success (or characteristically low success rates).
Furthermore, because the saliency models are expected to have varying degrees of accuracy, the confidence of a saliency model's prediction should necessarily factor into the assignment of α. By computing a confidence level for the minimum L1 difference weights by category, the confidence of each model is quantified and can be applied to Winnow. For the saliency model analysis implementation, the following modified Winnow was produced:

Algorithm 5 Confidence Level Winnow

Parameters: $G \in [0, 1]^T$ where T = the number of (chronological) predictions made by the expert, $D^T \in \mathbb{R}^T$, where D_i is the cumulative percentage of the predicted category's L1 difference, $C \in \mathbb{R}^T$ where $C_i = 2 * (D_i - 0.5)$, and $\forall i \in T, G_i \subset [0, 1]$

```
  $P \in [0,1]^T, P_1 = 0$
  $\alpha = 4^{Ci}$ if $P_i = 0, \alpha = 1/(8^{Ci})$ if $P_i = 1$
  for $i \in \mathbb{R}^T$ do
      if if $G_i \neq P_i$ : then
          for $\forall j \mid j \leq i$ do
              $\theta = j/2$
              $G_j = \alpha G_j$
          end for
      end if
      $P_{i+1} = 1$ if $\sum_{t=1}^{T}(G_t) \geq j/2$
      $P_{i+1} = 0$ otherwise
  end for
```

This algorithm possesses two qualities unique from the other Winnow models. First, the base of the positive multiplicative boost is half the base of the negative multiplicative boost. This means that for a model with perfect confidence ($\alpha = 1$), it would take two positive boosts to recover from one negative boost. This 2:1 ratio ensures that only algorithms with high success counts can recover from a false categorization. Second, the use of confidence level will benefit models that can produce high confidence levels. However, models that produce confident false categorizations will be demoted with a more aggressive multiplicative boost. A given model in this system may be able to correctly identify many images with low confidence, but its Winnow output might drop significantly with one highly confident bad categorization.

6 Results

The final seven pages of this report contain the graphs of the minimum L1 differences for each saliency model, the average L1 differences across the training image set for each model, and the Winnow2 output for the predicted category from multi-saliency model analysis.

6.1 Saliency Model Analysis:

One of the benefits of using a confidence level Winnow booster is it provides rich output data that can be mapped to a single curve. Along with model accuracy, the curve illustrates the individual confidence of the predictions that cause a reversal in the Winnow prediction. A high performing model under this analysis is one with highly-confident, successful predictions and low-confidence unsuccessful predictions. The antithesis, a model whose success are low confidence or has high confidence in unsuccessful predictions, should see a rapid downward convergence.
Starting with the minimum Hou graph, it is apparent that it has no capacity for correctly identifying car images. These elusive categories for experts are fairly common, especially without averaging. The obvious benefit of this output is Hou will never accidentally be selected for Car identification in the multi-saliency output, so the accuracy of the entire system increases from its removal. We also notice that the Hou model returns a mix of both high and low confidence values for its Cat categorizations, and we notice a pattern of success, failure, success in the ranges 76-80 and 92-96. This represents a significant part of the Winnow algorithm, where enough results have been recorded to predict two borderline consecutive failures even after two consecutive successes. The Hou wave predictions are not particularly confident, with the exception of the result at 104. That said, there are also a number of results from 20-48 that were incorrect, but with low-confidence. While not as exaggerated as the Cat data, this Hou Wave data is exactly what should be captured by this modified Winnow.
The Hou average results, on the other hand, are representative of a largely unsuccessful categorization model. Each category shows a few positive spikes in an otherwise unsuccessful classification span. Although some correct predictions (like the spike between 52 and 56 for Cat) are significant, they are not frequent enough to make the model trend up. The Hou wave data is somewhat interesting, in that it sees rapid fluctuation and high confidence in the beginning steps, but low confidence and a downward trend later in the span. This is a model that will slowly converge on zero after many trials, but its early Winnow boost will cause problems for the multi-model classifier until it sufficiently drops.
For its limitation with Car images, the Itti model's output is fantastic. The Cat predictions, while on the whole inaccurate, show a major confidence spike between 28 and 32. Furthermore, the Itti wave categorizations boast a success to failure ratio of 2:1 for significantly large confidence levels. Since the base of the false positive Winnow deduction is twice the false negative Winnow boost (for the modified Winnow in question), we expect the linear regression to be

negative for low-confidence models. As such, this Winnow algorithm favors the Itti Wave output because it can reliably produce a high confidence correct categorization within a reasonably limited span of trials.
The average Itti results are more erratic than their minimum counterpart, but it is one of the first models we see that can classify the Car images with some degree of accuracy. The Cat images go incorrectly classified for the majority of the trial, but the movement of the Wave and Car classifications is turbulent. The average result of the wave classification lies higher than that of the car, but the higher confidence car identifications give it a temporal edge.
The manifold classifications are both erratic and inaccurate, yet the confidence in results causes the output to trend up across the span. Each curve has at least one occurrence of a distinct success, success, failure spike pattern. The car curve repeats this frequently, while the cat curve only demonstrates this once. This low confidence boost makes $\sum_{t=1}^{T}(G_t) \geq \theta$ in the short run, and once the curve drops slightly below θ a high confidence prediction can significantly boost the result. The downward spike often surpasses the high confidence upward spike, but the total movement is still a net positive, resulting in long-timescale upward momentum. This momentum indicates a potential pitfall in the confidence-based Winnow, because it allows inaccurate (or at least below average) models to accrue greater weighted influence in the multi-model categorization.
Manifold average produces results that resemble the average models, with the notable exception of the car category. For manifold average car, we see that the results trend upwards between spikes for two significantly long stretches (32-68 and 72-82). Previous models had trended down between spikes, and relied on false negative predictions and high confidence levels to propel their Winnow output upwards. In this case, however, manifold car seems to be consistently increasing on its own, and the positive spikes act as a true "boost" for an otherwise trending-up prediction. Early correct predictions for all three categories artificially increased the output for a brief span, but the plateaus and subsequent downward spikes indicate that the initial spike was a product of luck and not model accuracy.

6.2 Multi-Model Winnow2 Analysis:

Winnow2 does not and is not expected to converge on any one value, due to the fixed α for upward and downward spikes. Instead, it fluctuates between two semi-fixed ranges, the upper for subsequent successes and the lower for subsequent errors. As such, the algorithm output in the short run is a measure of consistency. Bear in mind that the category prediction comes from the saliency model with the maximum magnitude Winnow output for the predicted category, so this Winnow2 output is therefore a measure of the consistency with which the correct model is chosen.
This sample output begins with a random success from a set of completely un-weighted saliency models. Unsurprisingly, the results decline for twelve iterations, dropping well below 0.25. It only takes three correct predictions after this point to boost the predictions back to almost 1, and by image 32 convergence begins in earnest.
This two point convergence initially averaged slightly above the 0.5 mark, but by the end of the trial it was slightly below. This is only readily apparent running a linear regression on the set. Unfortunately, the unmodified curve does not give much additional insight into how this model functions. If more images were tested, the two points would eventually converge on some value. The average in this sample output was 0.41, and the upward linear regression line indicates the convergence point might be higher than that.

At least with the current data, the outcome of the multi-saliency predictions perform slightly, but substantially better than a random guess across the three categories. The fast convergence time for Winnow2 means that the order of successes and failures matters, and for future tests running a number of randomized iterations of the entire testing set may produce interesting Winnow2 results.

7 Conclusion:

7.1 Saliency Models:

As predicted, there exists no one-size-fits-all model for realtime saliency computation. Some models, like the graph-based manifold saliency, are decent general purpose predictors, though it does not excel at any one category. Others, like Itti and Hou saliency, do a good job of accurately predicting individual categories, but have absolutely no ability to pick others. However, the models are diverse enough in their error and, more importantly, in their correct predictions that combining their efforts produces a better result than their individual output.
That said, the multi-saliency prediction does not function to the degree that it should in terms of accuracy. Part of this is due to the selection of a single model for the prediction, as opposed to a weighted average. The problem with an average is it requires a norm for the output span of the Winnow algorithm, which usually involves division by the maximum of the set. For many of the model "average" outputs, they record an artificially large spike at the beginning of the trial that is much higher than the result of any later boost. While that quality might not inherently produce bad results, early attempts at this addition proved unsuccessful.

7.2 Winnow:

One of the interesting traits of the Winnow framework is slight variations on α can change the function of the algorithm drastically. Winnow1 converges on a binary result incredibly fast, and it can be easily extended to work in high dimensional frameworks. Winnow2, for our purposes, will eventually converge on a value between 0 and 1, and can act as a reasonable measure of accuracy for a system.
The confidence level Winnow is less stable than the previous two Winnow models, but it can be highly beneficial for short bursts of similar inputs. In earlier experiments during implementation, a confidence level Winnow was run against the saliency models, without specifying category. At this point, the data set had not been randomized, so often large spans of the testing images would be heterogeneous in category. The confidence level Winnow could rapidly upgrade/downgrade the predicted efficacy of a model as the stream of inputs occurred, which is a kind of versatility not possible with Winnow 1 or 2. Winnow seems well suited to this extensibility, because as long as the Winnow result predicts an incorrect categorization, the multiplicative update by α will hold. This rapid transformation makes winnow well suited to somewhat volatile data sets and online learning scenarios.

7.3 Future Extensions:

While most of the key components of this categorization program came to fruition, a few components could be added to make it more effective.

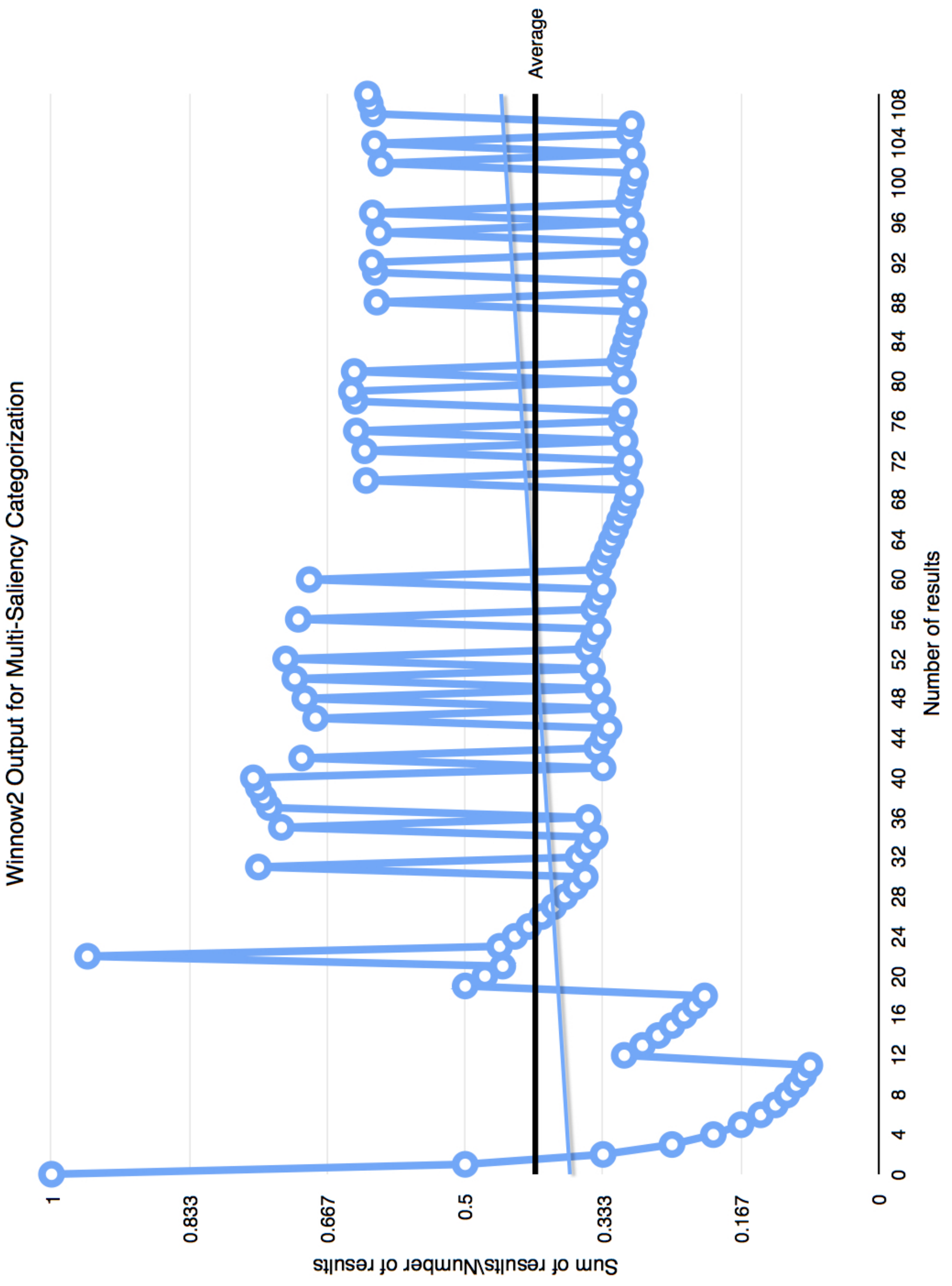
Training images for this project were not created equal, and the presence of bad training images in the set may have resulted in inaccurate classifications or poorly weighted model averages. As such, it would make sense to adaptively prune models that consistently result in large L1 differences for testing images. Were the size of the training data sets larger, a confidence interval could be calculated for the L1 difference of individual training saliencies, and significantly large saliency differences could be purged from the set. That said, some of the training data sets were smaller than the testing image sets, so the training data set might be empty by the end of the iteration.

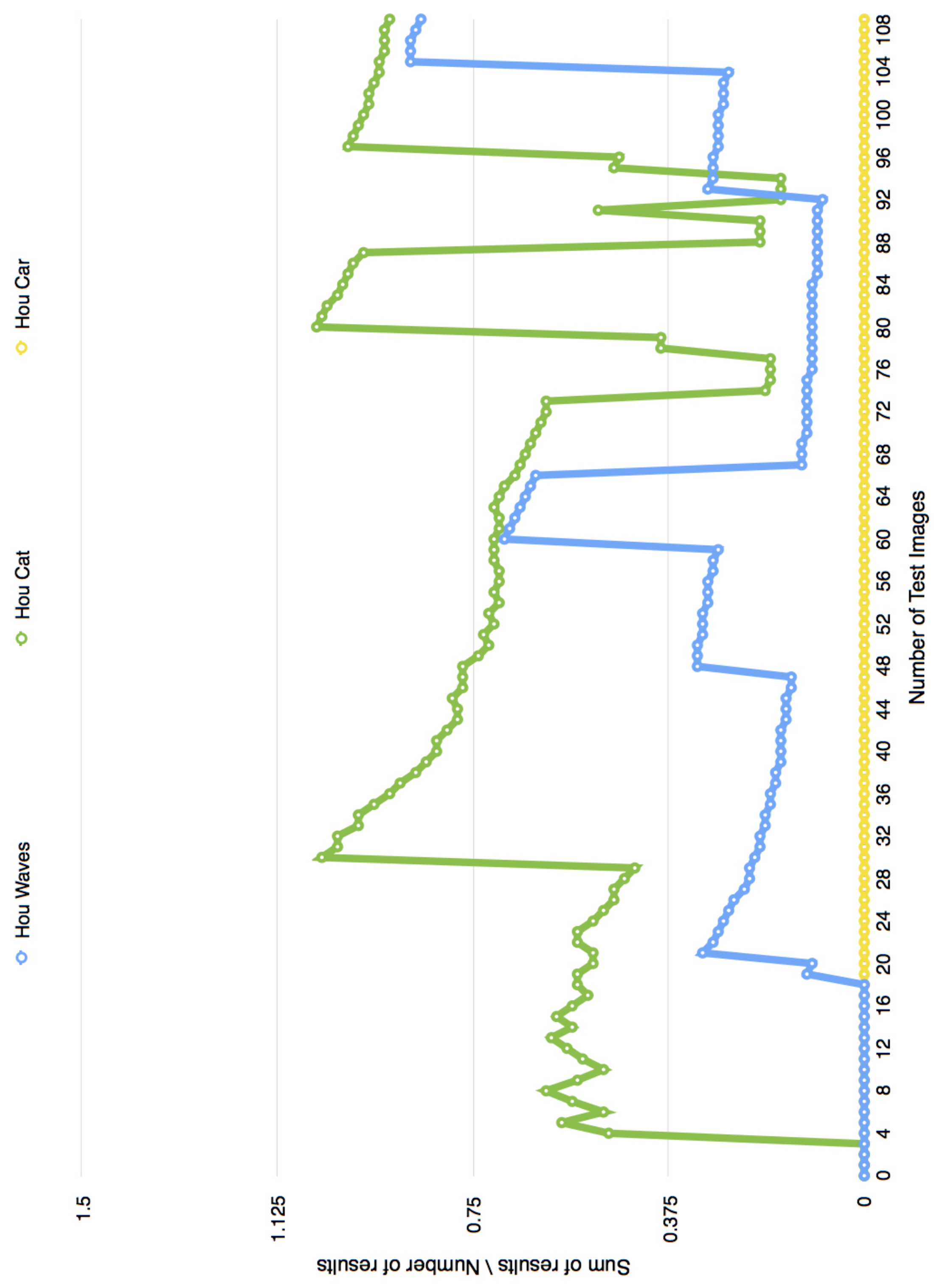
Another extension could be the automatic addition of correctly identified image saliencies to the appropriate training saliency category. This is especially feasible and desirable for a real world application, but would cause inaccurate results for fixed data sets.

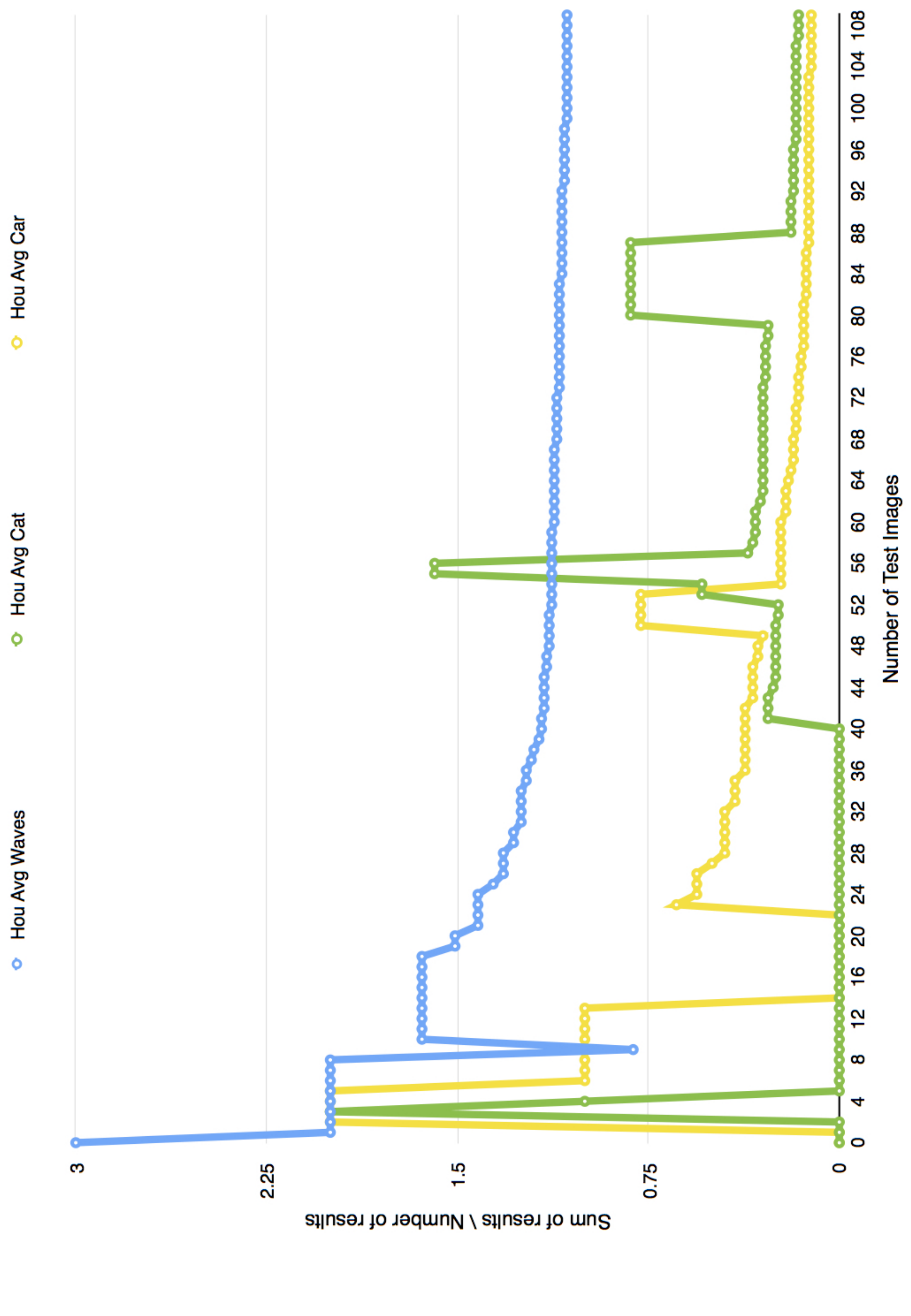
One final, and more complex, addition would be a saliency map convolution using the confidence level Winnow output. This convolution could produce a self contained salient image that would replace the multi-saliency estimate (selecting the classification of the model with the highest Winnow output). Each model would produce its own saliency and, given the Winnow outputs from the trial before, the individual models would be weighted and convolved to produce this new estimate. The major caveat to such a system would be the recalculation of each training saliency with these parameters at every test image iteration. While not significant enough to be considered computationally expensive, this increase in runtime would prevent this model from scaling as well to larger data sets.

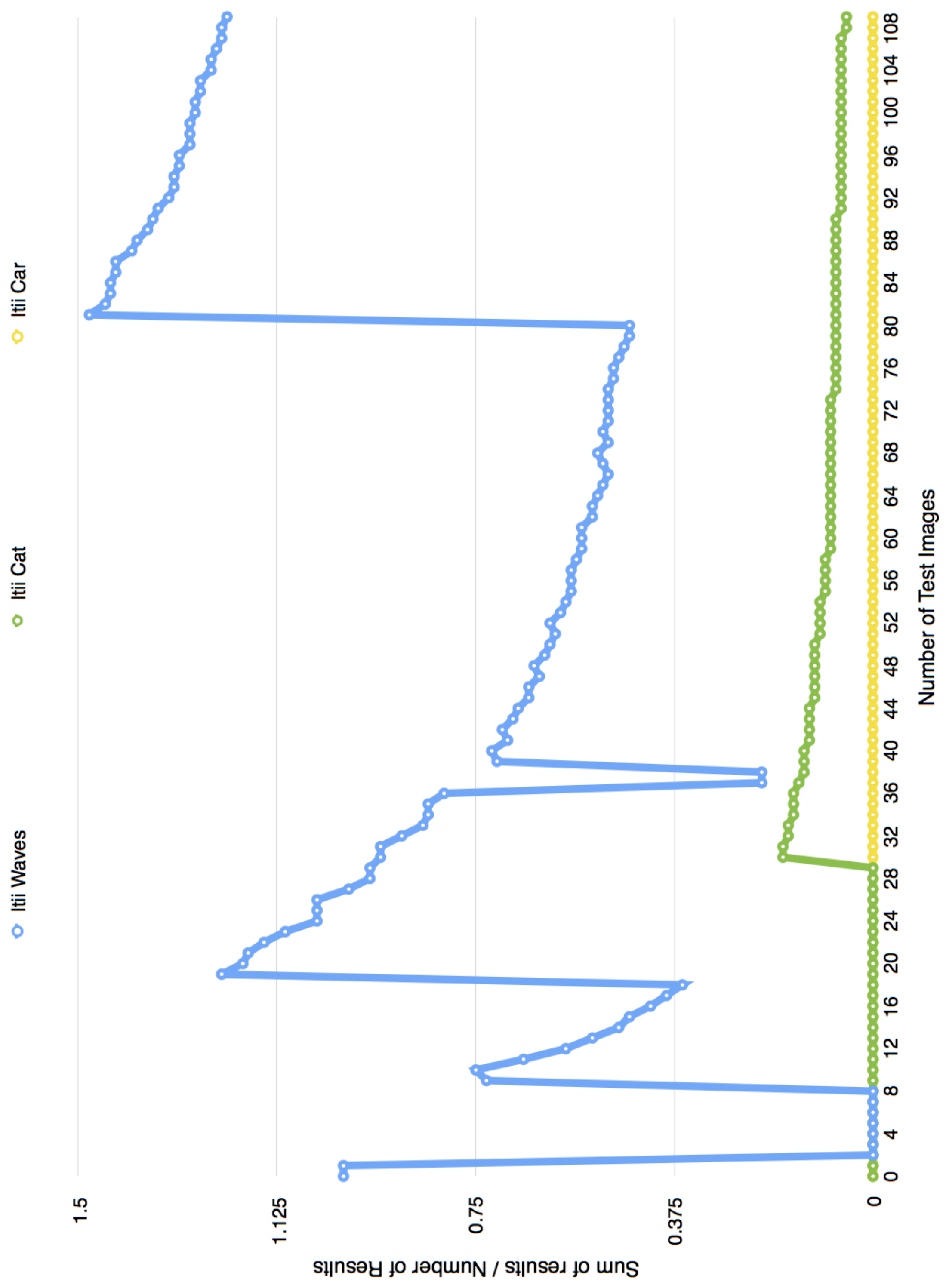
7.4 Citations:

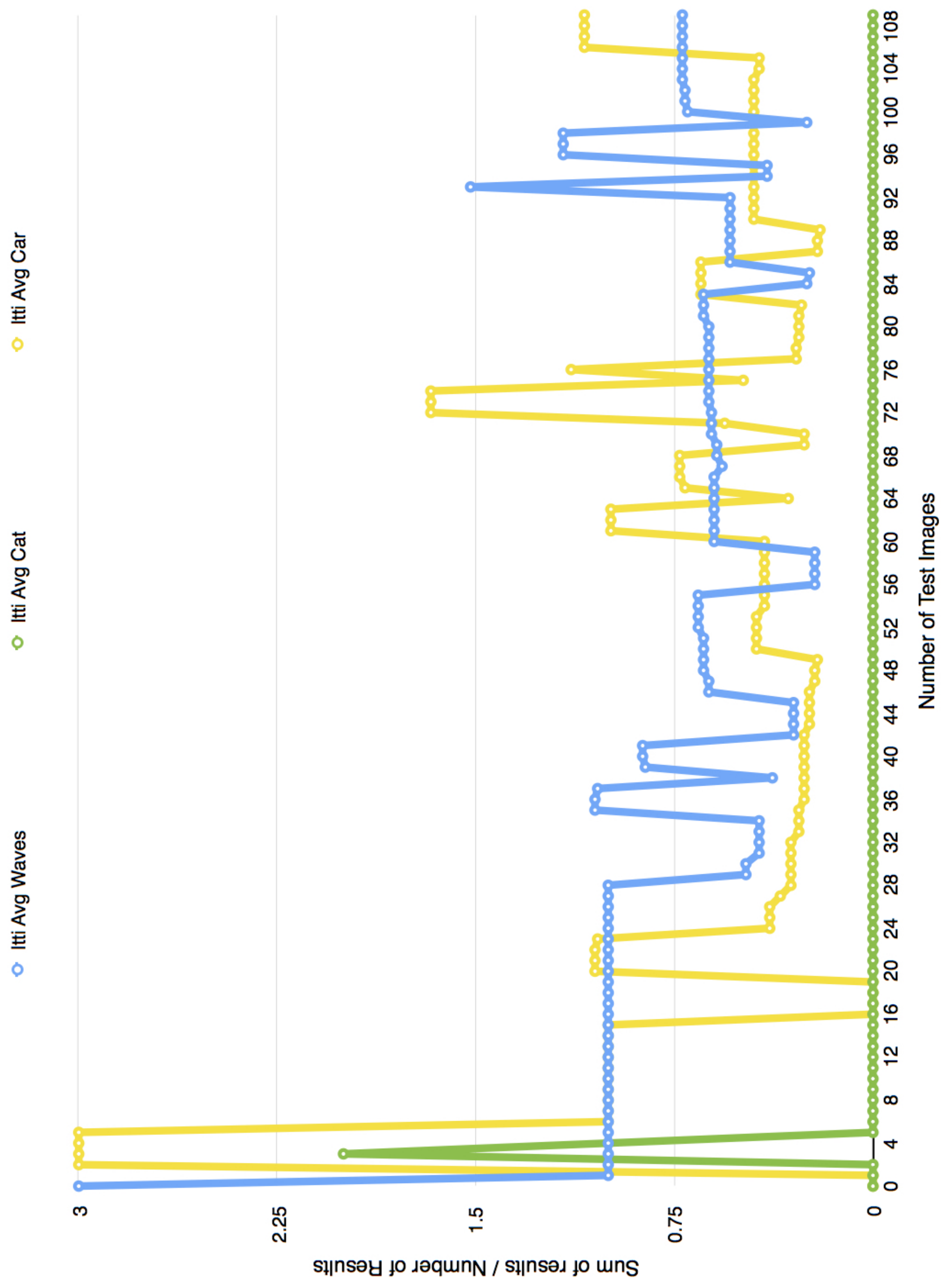
- A. Kimura, "pySaliencyMap", (2016). GitHub repository, `https://github.com/akisato-/pySaliencyMap`
- C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency de- tection via graph-based manifold ranking. In CVPR, 2013.
- Google, "multibox", (2015). GitHub repository, `https://github.com/google/multibox`
- L. Itti, C. Koch (1999). "A saliency-based search mechanism for overt and covert shifts of visual attention". Computation and Neural Systems Program, Division of Biology, California Institute of Technology.
- L. Itti, C. Koch, E. Niebur, A Model of Saliency-Based Visual Attention for Rapid Scene Analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 11, pp. 1254-1259, Nov 1998.
- M. Beyeler, "opencv-python-blueprints", (2015). GitHub repository, `https://github.com/mbeyeler/opencv-python-blueprints/blob/master/chapter5/saliency.py`
- N. Littlestone (1989). "Mistake bounds and logarithmic linear-threshold learning algorithms". Technical report UCSC-CRL-89-11, University of California, Santa Cruz. "Winnow (algorithm)." Wikipedia. Wikimedia Foundation, n.d. Web. 15 Dec. 2016.
- X. Hou and L. Zhang (2007). Saliency Detection: A Spectral Residual Approach. IEEE Transactions on Computer Vision and Pattern Recognition (CVPR), p.1-8. doi: 10.1109/CVPR.2007.38326
- X. Ruan, "mr_saliency", (2015). GitHub repository, `https://github.com/ruanxiang/mr_saliency`

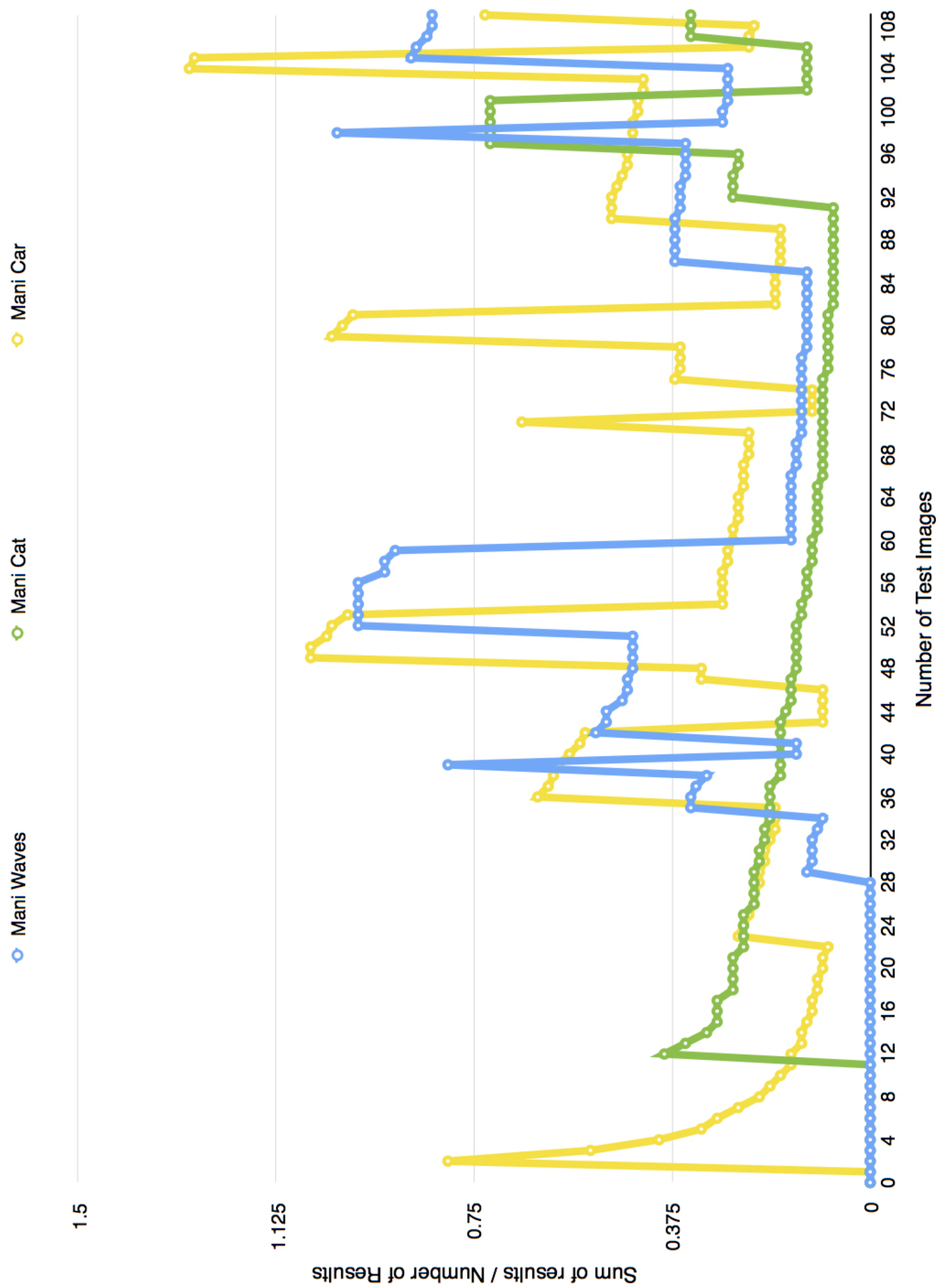

Winnow2 Output for Multi-Saliency Categorization
Sum of results\Number of results
1
0.833
0.667
0.5
0.333
0.167
0
Number of results
0
4
8
12
16
20
24
28
32
36
40
44
48
52
56
60
64
68
72
76
80
84
88
92
96
100
104
108
Average

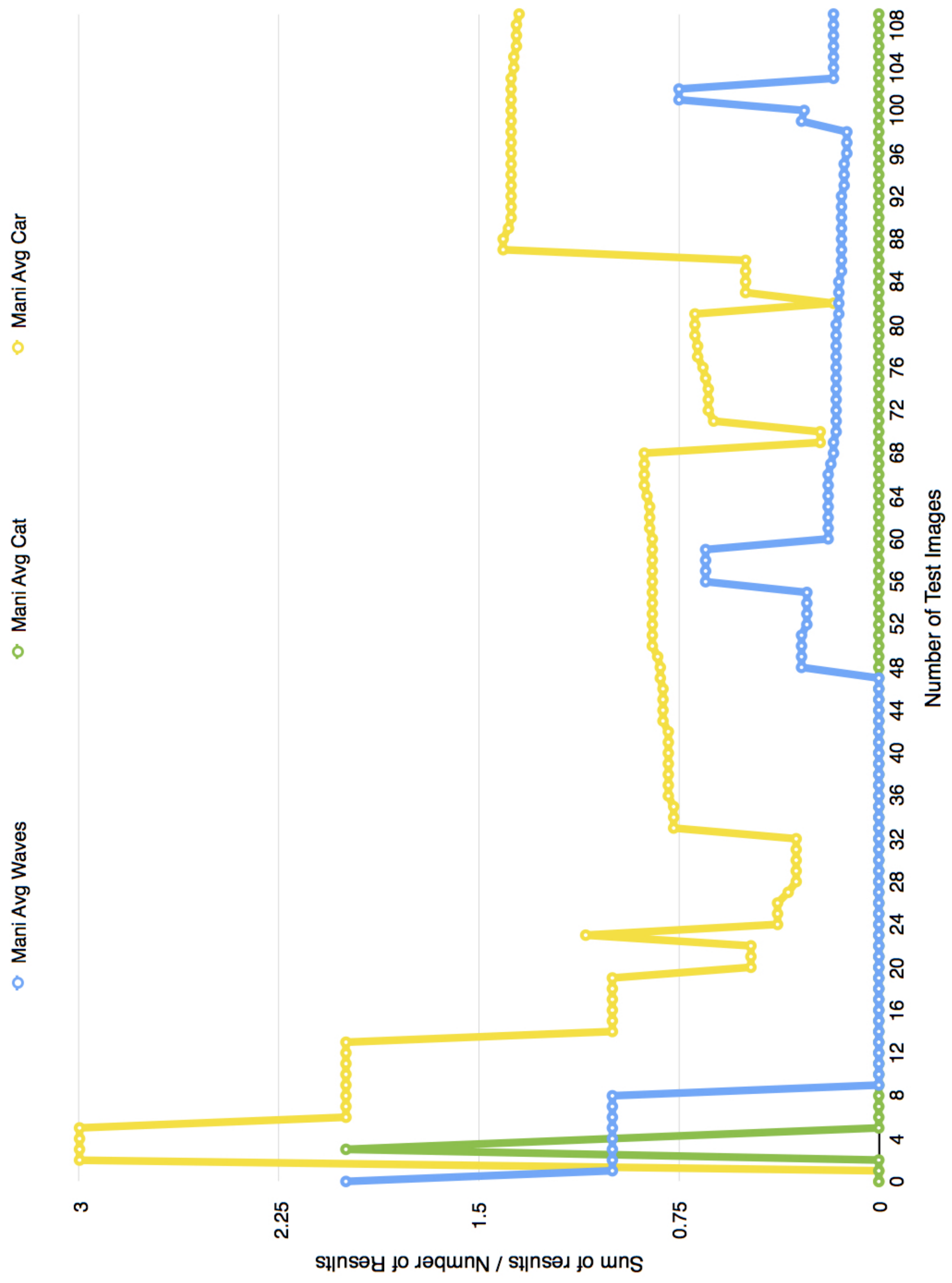

Hou Waves
Hou Cat
Hou Car
Sum of results \ Number of results
1.5
1.125
0.75
0.375
0
0
4
8
12
16
20
24
28
32
36
40
44
48
52
56
60
64
68
72
76
80
84
88
92
96
100
104
108
Number of Test Images


Hou Avg Waves
Hou Avg Cat
Hou Avg Car
Sum of results \ Number of results
Number of Test Images
0
0.75
1.5
2.25
3
0
4
8
12
16
20
24
28
32
36
40
44
48
52
56
60
64
68
72
76
80
84
88
92
96
100
104
108


Itii Waves
Itii Cat
Itii Car
Sum of results / Number of Results
Number of Test Images
1.5
1.125
0.75
0.375
0
0
4
8
12
16
20
24
28
32
36
40
44
48
52
56
60
64
68
72
76
80
84
88
92
96
100
104
108


Itti Avg Waves
Itti Avg Cat
Itti Avg Car
Sum of results / Number of Results
Number of Test Images
3
2.25
1.5
0.75
0
0
4
8
12
16
20
24
28
32
36
40
44
48
52
56
60
64
68
72
76
80
84
88
92
96
100
104
108


Mani Waves
Mani Cat
Mani Car
Sum of results / Number of Results
0
0.375
0.75
1.125
1.5
Number of Test Images
0
4
8
12
16
20
24
28
32
36
40
44
48
52
56
60
64
68
72
76
80
84
88
92
96
100
104
108


Mani Avg Waves
Mani Avg Cat
Mani Avg Car
Sum of results / Number of Results
Number of Test Images
0
0.75
1.5
2.25
3
0
4
8
12
16
20
24
28
32
36
40
44
48
52
56
60
64
68
72
76
80
84
88
92
96
100
104
108