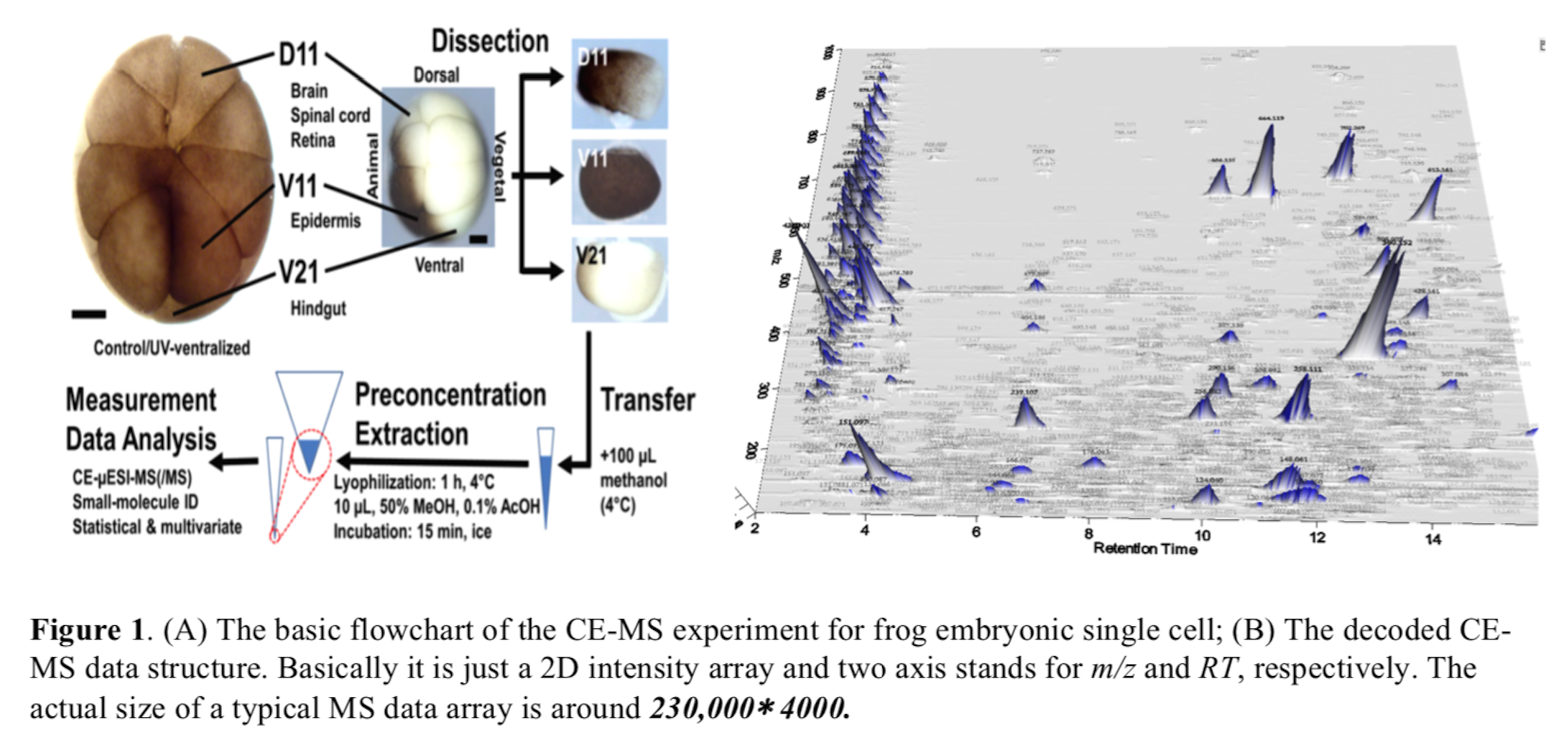
**Machine Learning based detection of metabolic signal in signal in single cell mass spectrometry**

Jifei He, Wanding Zhang

May 15, 2018

**1. Background**

Spatiotemporal characterization of molecular expression during embryonic development is critical for understanding how cells become different and give rise to distinct tissues and organs. Technological advances enabled the measurement of RNAs and proteins in single cells of embryos, but there is very limited information on small molecules, metabolites, that are the ultimate indication of the physiological state. To fill this knowledge gap, a single-cell technology was developed and used to find that embryonic cells that give rise to different tissues and organs have characteristically different metabolic signatures that affect cell fate. The capillary electrophoresis (CE) and mass spectrometry (MS) were combined to reveal the abundance of each small molecule (range from 25 Da to 550 Da) within a single cell with high resolution (up to 0.1 mDa). Basically CE was used to separate different molecules in a tiny capillary (different by Retention Time, or RT) and then MS evaluated the composition (different by mass to charge ratio, or m/z), with a certain scanning frequency. Thus, the MS data from experiment is a 2D intensity array with one dimension standing for m/z while the other for RT.

**2. Methods and Findings**

**2.1 Machine Learning models -** Works done in previous stages have explored different methods for signal extraction before calling for a search of effective features in full 2D intensity profile using machine learning (ML), in particular, deep learning (DL) techniques. Treating the intensity profile as 2D image allows for further utilizing deep convolutional neural networks (CNN) for pattern recognition and object detection in image analysis. Give the fact that deep learning has achieved great success in image segmentation and object detection with wide applications in astrophysics, biology, and many other areas with superior performance, we set off our expedition by first exploring ML models and several famous CNN models in computer vision history and comparing their performance in the metabolomics datasets, providing a reference for the application of deep learning in other research area.

**1) Training dataset -** Large dataset is necessary to train most of ML and DL models such as the ones used here. Based on both initial screening and human inspection of a large set of 2D images of potential signals from animals, a total of 4546 potential signal images are labeled resulting in 1464 true signals and 3082 false signals. These 4546 images in total constitute our training pool for supervising learning. To even out signal-to-noise ratio (SNR), a method is adopted to randomly select an equal number of images from both the true and the false labels, in that, we often start our models with 2928 images as our training set.

**2) Model training**. With the training dataset obtained above, the neural network is trained by adjusting the weights using backpropagation and gradient descent algorithms. We use the cross entropy as our loss function for minimization and adjusting the weights. The training optimizer used is ADAM in TensorFlow with a learning rate of . Other parameters are mostly set as default in TensorFlow. The model is trained till the final training accuracy converges with little fluctuation. From our hype-parameter tuning experimentation, 5000 or so training steps turn out to be sufficient, and the process takes about 5 minutes on a NVIDIA K20 GPU. Training the model on a standard computer without GPU should take no more than 30 minutes.

**3) Evaluation of signal images**. After the neural network machine is trained to achieve a reasonable accuracy, bounding box of signal images to be evaluated are fed to the machine to be given a probability score for being a true signal. This probability serves as the prediction confidence score, i.e., the higher the probability, the more likely that the image represents a true signal.

**4) Findings and Conclusions**

* **Single Layer Neural Network**

|  |  |
| --- | --- |
| Loss function: softmax\_cross\_entropy\_with\_logits  Optimizer: Adam Optimizer  Learning rate: 1e-3  Batch size: 4,000  Epoch: 80,000  **Accuracy: 0.871701** | **Single Layer Neural Network Weight Plots**  Red - positive reaction  Blue - negative reaction |

The weight plots shown above depict the machine’s general understanding of the false signals (weight: 0) and the true signals (weight: 1) respectively. The plots show a distinctive boundary between two classes with a center round shape and the rest surrounding it. Before quickly drawing a conclusion, we want to enhance this understanding that the machine differentiates the true from the false mainly by the center part of the images.

**Exploring Truncated Weight and Its Effect on Accuracy**

|  |  |
| --- | --- |
| Set weights[0:200] =0  Epoch = 10  **Accuracy = 0.755404** |  |
| Set weights[200:600] =0  Epoch = 10  **Accuracy = 0.679181** |  |

In that, we can conclude that the center part of the images are the essential for pattern recognition and images classification.

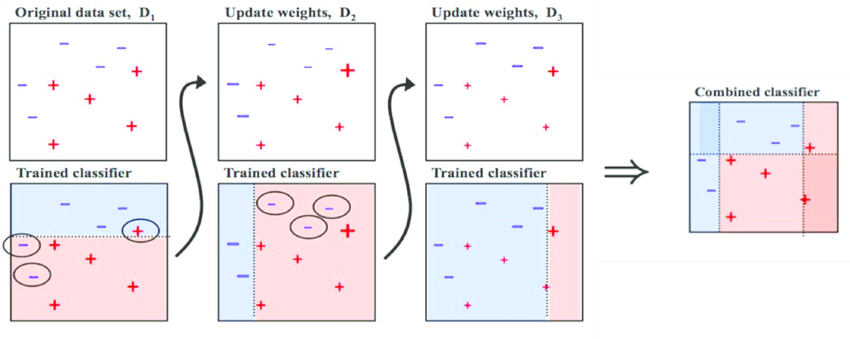
* **Multi-layer Neural Network**

|  |  |  |
| --- | --- | --- |
| Input : 800  Hidden layer 1 : 256 neurons  Hidden layer 2 : 256 neurons  Batch size: 100  Epoch: 80,001  **Accuracy: 0.8673** | **Accuracy Plot** | **Cost Plot** |

Given that all the labels in this datasets are all hand-labeled, it’s common that some of the images are not black-and-white distinctive and thus hard to classify. We want to pick out those images in the grey area or those are ambiguous to label. In that we developed a

method inspired by the known Adaboost mechanism to pick out only the ambiguous

images. Specifically, we started training with a traditional multi-layer neural network for a small epoch size (about 3000) before putting back the correctly labeled images as a training standard to continue training the rest of the images. By that, we forced the machine to intensively remember the features of images that are easy to correctly predicted. A small batch size (about 100) is adopted to expedite the process. After about 100 epoches, the machine was able to correctly classify 2660 images and left only 268 images in question. One thing to highlight about this method is that we set a strict prediction ratio to it. Since we have only two labels (the true and the false respectively), a pair of possibilities is generated for each one predict/image. For example, a predict for image #10 could be [0.64 0.36] as 64% to be labeled as the false and 36% as the true. Traditionally, the machine will classify/label image #10 as the false as the possibility favors more on one class. However, we considered that this is not a very strong and absolute classification. So we then set the possibility threshold to 0.9 and collect the correctly labeled images that can only reach this high possibility. In that, image #10 is no longer predicted as a correct prediction.



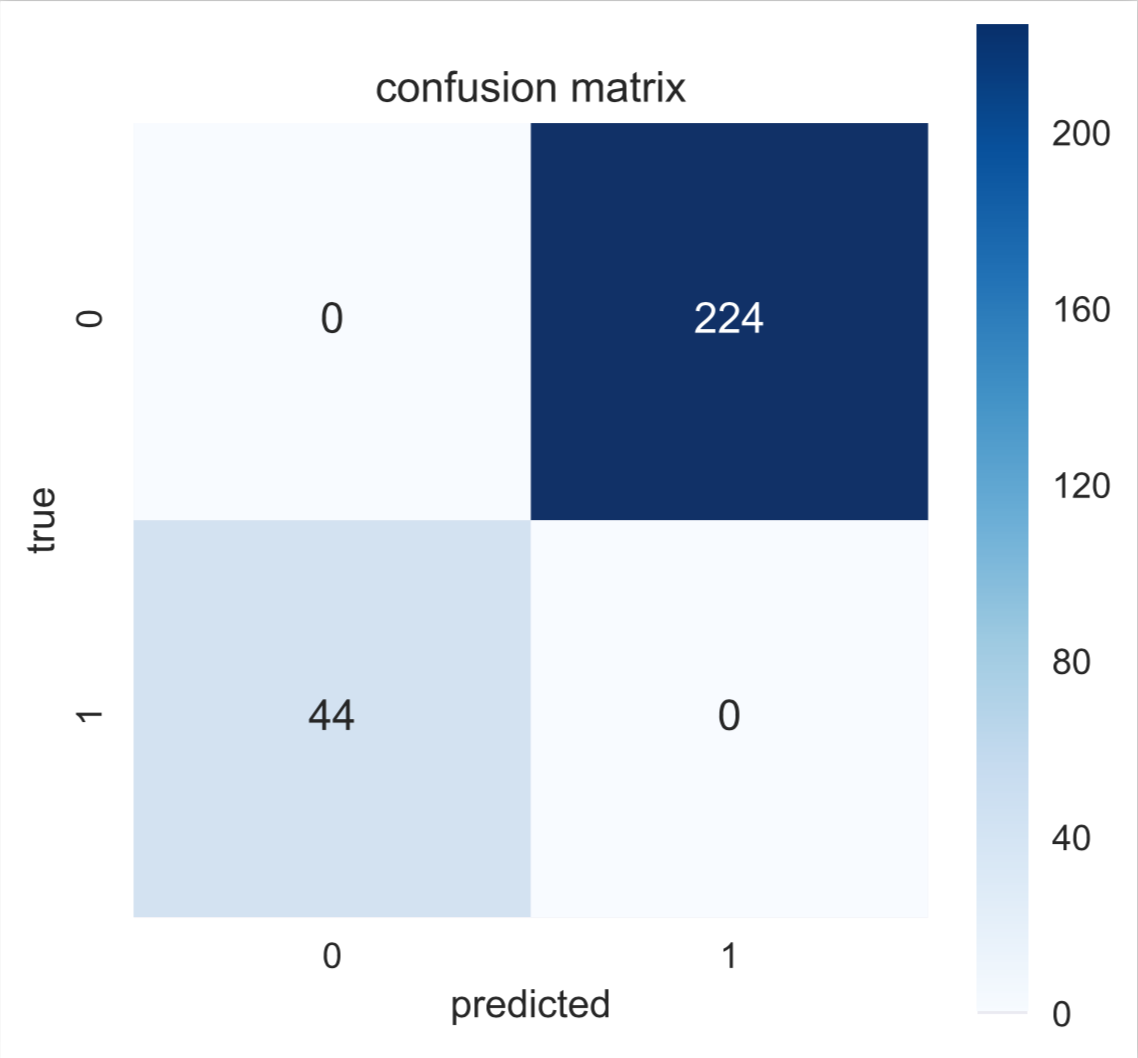
**AdaBoost Classifier**

Source: https://www.researchgate.net/figure/Training-of-an-AdaBoost-classifier-The-first-classifier-trains-on-unweighted-data-then\_fig3\_306054843

We can see that the machine can successfully label images with distinctive and explicit features while picking out those that are ambiguous to classify. With this machine, we would be able to save a lot of time in future encountering similar unlabeled metabolic signal datas.

|  |  |
| --- | --- |
| **Leart in the early stage** | **Unable to learn after 100 epochs** |
|  |  |

**Results for Our Method**

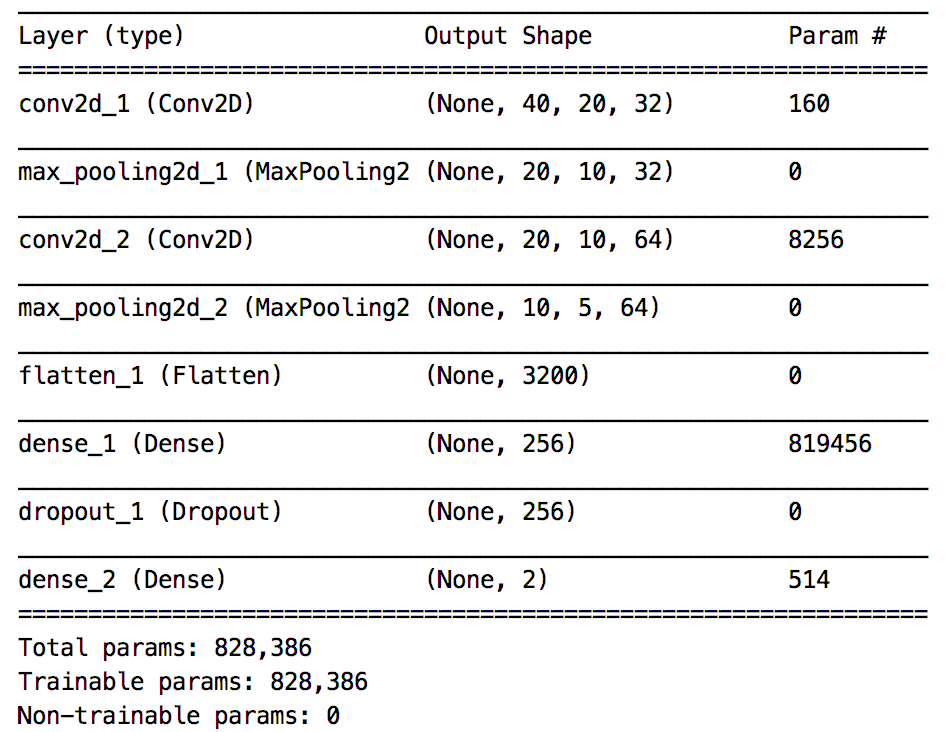


**Confusion Matrix for False:True Signals**

In the 268 images that the machine was unable to label, 224 were hand-labeled as the false and 44 were the true.

* **Convolutional Neural Network**

|  |  |  |
| --- | --- | --- |
| Loss function: softmax\_cross\_entropy\_with\_logits  Optimizer: Adam Optimizer  learning rate: 2e-5  batch size: 100  epoch: 3,001  Keep rate: 0.8  **Accuracy: 0.875057** | **Accuracy Plot** | **Cost Plot** |



Comparing with Single and Multi-layer Neural Network, a traditional Convolutional Neural Network model did not show a prominent advantage in recognizing patterns in metabolic signals.

**Convolutional Neural Network Model Architecture**

**2.2 Similarity**

**1) Training dataset -** At this point, we moved from animal datasets forward to a set of human liver metabolic signal datas of 299\*4 potential unlabeled signal images. The four sets of 299 images stand for four biological samples while each 299 metabolite signals ought to be similar in time and m/z.

We were exploring a method to differentiate and label those signals by comparing the similarity rate. After browsing, examining and comparing images, we proposed a hypothesis that a true signal look alike while a false signal look relatively different in shape and intensities across four biological samples.

We calculated the Cosine similarity between every pair of two images in the four biological samples by their inner products. The results show that images appear to be true signals generally have high similarity percentage (above 90%) while the false signals fell low.

|  |  |
| --- | --- |
| **High Pair Similarity** | **Low Pair Similarity** |
|  |  |

**Pair Similarity**

**2.3 Transfer learning**

Use pre-trained models as the starting point on computer vision and natural language processing tasks is becoming a popular approach in deep learning given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems (Brownlee, J. 2017). We also wanted to introduce a deep learning model pre-trained for the large and challenging image classification task on which we were working.

**1) VGG Model** - The pre-trained model we used is **Oxford VGG Model**. VGG is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition” . The model achieves 92.7% top-5 test accuracy in ImageNet , which is a dataset of over 14 million images belonging to 1000 classes (Frossard, D. 2016).

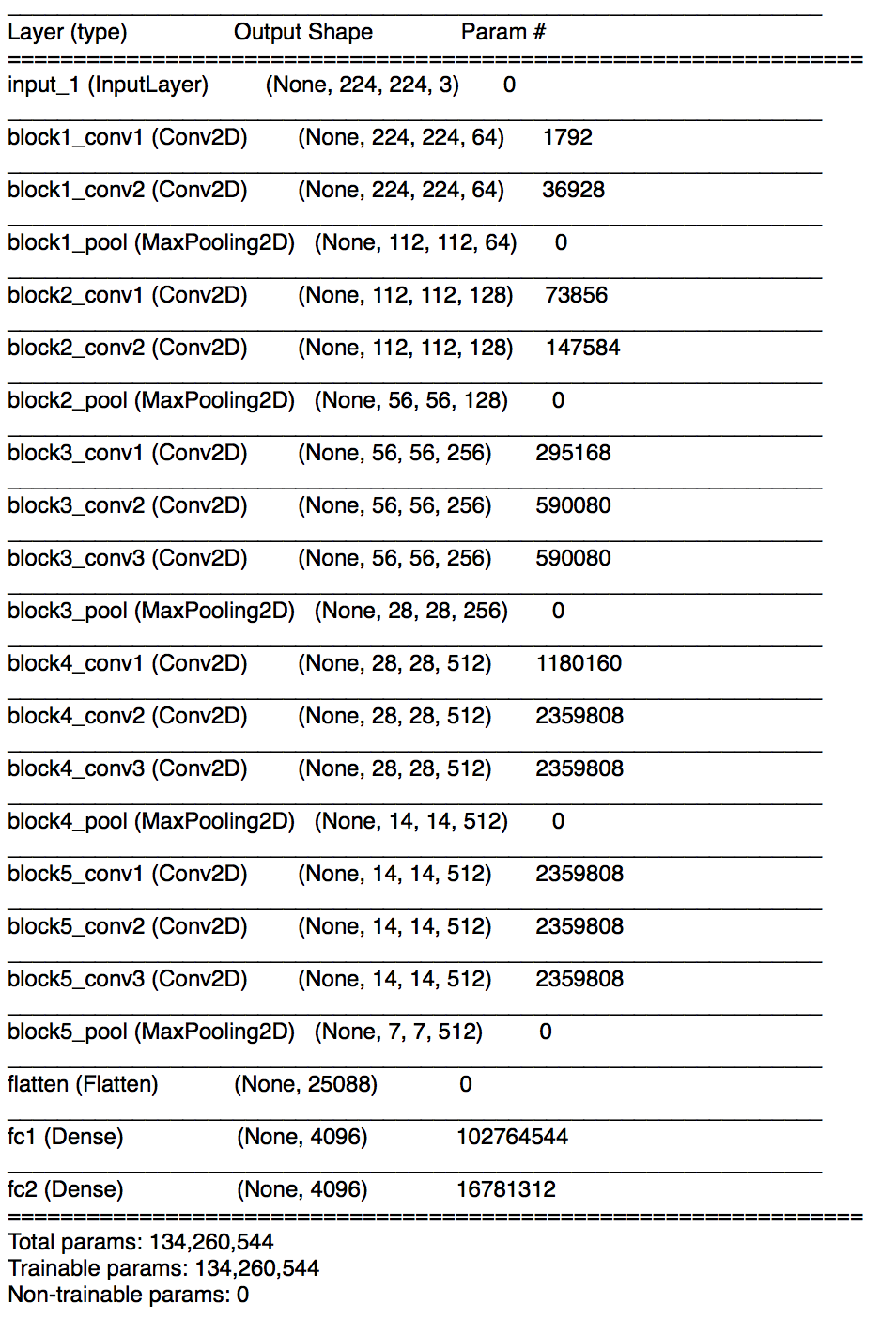
Instead of running through the model to the end and getting a class prediction, we truncated the model to the fc2 layer (right before the prediction) and generate a 4096 vector for each image. The model summary can be found below.

We tested VGG model on our previous animal data but the results are inexplicable and indistinguishable. It can be explained that the animal image is originally 800 points before being trained in VGG and it was scaled to 4096 after training, thus the features that VGG picked out are scattered apart.

|  |  |  |
| --- | --- | --- |
| **Animal 40\*20\*1** | **Reshape Animal 224\*224\*3** | **VGG fc2 64\*64\*3** |
|  |  |  |

**Oxford VGG Model Preview Images in Animal Datas**

(Resorce: <https://machinelearningmastery.com/transfer-learning-for-deep-learning/>)



**VGG Model Architecture**

**2) Image Transformation**

* **Shift**

|  |  |
| --- | --- |
|  |  |

When we compare similarities between original images and images that shifted left, similarities that calculated with VGG processed data are obviously higher.

* **Rotation**

|  |  |
| --- | --- |
|  |  |
|  |  |

When we rotate the image 180°, in most cases, with VGG processing, the similarities between transformed images and original images are higher.

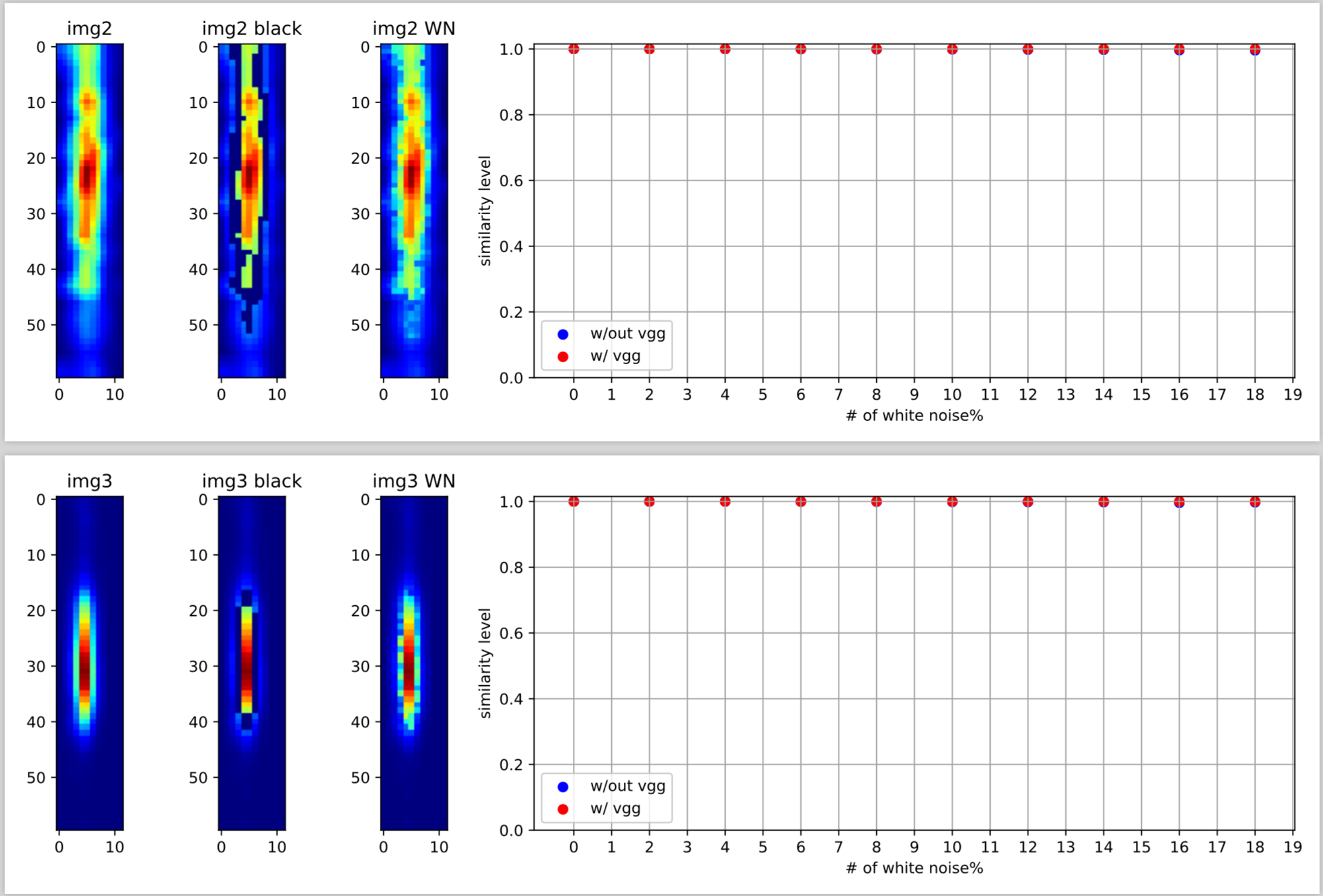
* **White Noise**

1. **Add white noise to the whole image**

|  |  |
| --- | --- |
|  |  |

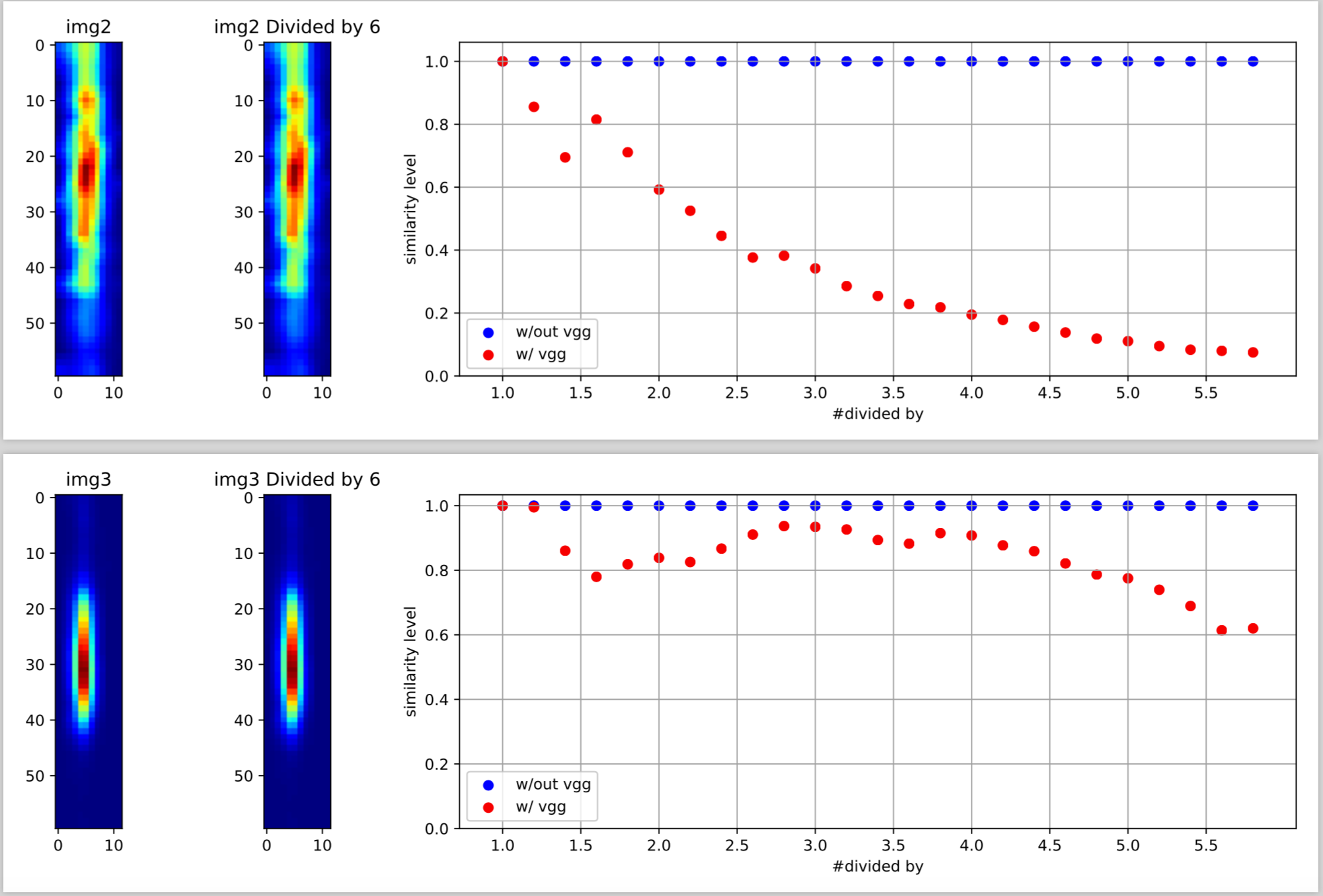
VGG is very sensitive to white noise.

1. **Only add white noise to the border**

****

We scale the value of pixels to the range of (0, 255), and add noise to the pixels in the range of (60, 130), which is the border of the bright spot. From the plots we can see that, with or without VGG, white noise almost does not affect the similarity between transformed images and original images.

* **Scale**

****

We divided the pixels value(0, 255) by a number in the range of (0, 6), and calculated similarities between original images and transformed images. VGG is sensitive to scaling.

**2.4 Entropy**

**1) Training dataset -** The previous liver datasets of four biological samples and 1 time belongs to a larger datasets comprised of eight biological samples and 10 times. The previous Similarity method can be used to label signals but it is very time consuming. 80 images per signal would generate 3160 pairs of images and thus 3160 similarity values. Given it, we want to find a better and time-efficient method to calculate similarity.

|  |  |
| --- | --- |
| **Previous Liver Dataset** | **Full Liver Dataset** |
| Times: **30m**  Groups: **C1, C2, C3, C4** | Times: 3-6s, 6m, **30m**, 1h, 2h, 4h, 8h, 18h, 24h, 48h  Groups: **C1, C2, C3, C4**, T1, T2, T3, T4 |

C stands for Control and T stands for Test Group.

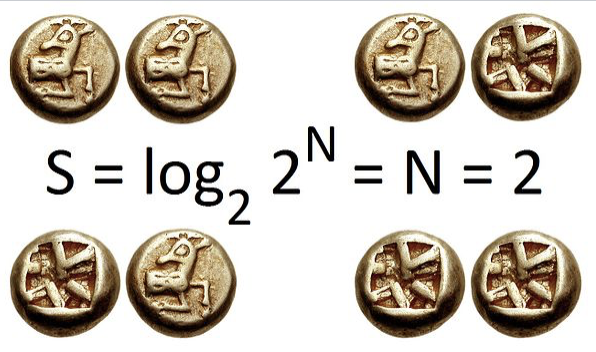
|  |  |  |
| --- | --- | --- |
| **Full liver dataset** | | |
|  |  |  |

**2) VGG Predictions**

|  |
| --- |
|  |
|  |

By running through VGG to the end and classified images to one of the 1000 classes, we left the top three classes that VGG predicted. It is interesting to find that a true signal generally have a relatively high possibility in its top one class while a false signal have relatively low possibilities across all three classes. Given that, entropy could be a method to distinguish true signals from false, in other word, a good substitute method to the Similarity method.

After training by VGG, each image can generate a 1000 points vector. The function we adopted to calculate entropy is shown below. A entropy value will be produced for each image. The image with a small entropy is generally considered as a true signal and VGG has more confidence (or possibility) that the image belongs to one class.



**Entropy Calculation**

Source: <https://en.wikipedia.org/wiki/Entropy_(information_theory)>

We first examined the entropy values across different time ranges and the results show a relatively steady level. 



The scatter plot between entropy and the maximum intensity of the images indicates a negative correlation that entropy tends to be small in images with greater maximum intensity.

|  |  |
| --- | --- |
| **Entropy vs. Maximum Intensity Scatter Plot for C and T groups respectively** | |
| **Control** | **Test** |
|  |  |

**3. Summary**

As time brought us to the end of this semester, our learning and researching in this project has not yet come to a close. We still need to reconcile datas of our work and continue contributing to the paper of this project. In this project, we explored a new area in pattern recognition which is neither of our main fields of study before this project. We gained valuable experience. And we were introduced some very inspiring methods in problem solving. In return, we contributed our part in the process and made interesting findings which we hope can be facilitate better recognizing and classifying metabolic signals in the field, and maybe more.

**4. References**

Amato,F. et al. (2013) Artificial neural networks in medical diagnosis. 47–58.

Matsuyama,E. et al. (2013) A Modified Undecimated Discrete Wavelet Transform Based Approach to Mammographic Image Denoising. 748–758.

Brownlee, J. (2017, December 20). A Gentle Introduction to Transfer Learning for Deep Learning [Web log post]. Retrieved May 18, 2018, from https://machinelearningmastery.com/transfer-learning-for-deep-learning/

Frossard, D. (2016, June 17). VGG in TensorFlow [Web log post]. Retrieved March 18, 2018, from https://www.cs.toronto.edu/~frossard/post/VGG16/