Final Project Report: Classification of Type 1a Supernovae

Abel Yagubyan: 3034188018

CS294-082: Experimental Design for Machine Learning on Multimedia Data

Professor Gerald Friedland

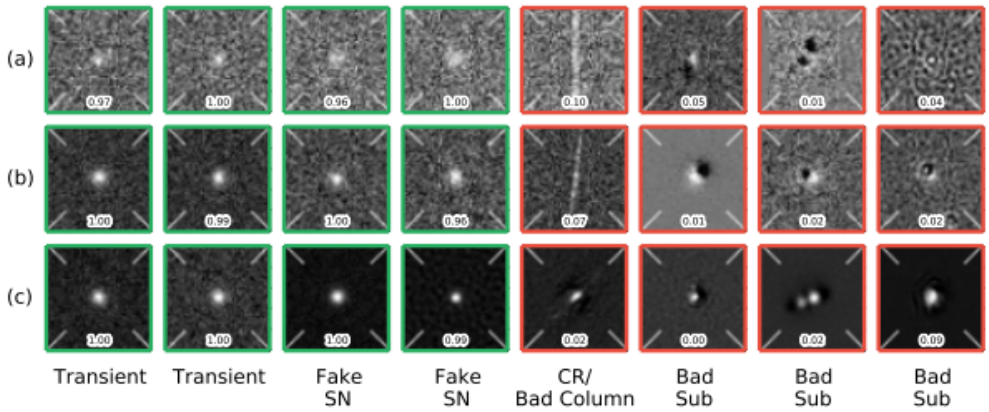
December 20, 2021

Abstract:

With the revolutionary introduction of Neural Networks within modern world issues, there have been many useful applications of it in different fields. Specifically to this report, within the field of image processing, Convolutional Neural Networks have been modern staples to exploiting localities at different granularities and modeling features within images, thereby helping to deduce useful information from them. In particular, we used two separate Machine Learning models with the help of a Random Forest model [1] and a CNN model [2] to actively attempt to classify astronomical images of Supernovae from the Dark Energy Survey [3] and see what fits best. Please note that due to the better accuracy of the former model, I investigated the particular CNN model until the third question, then I just focused on the Random Forest model due to its higher accuracy.

Introduction:

The dataset used is from the Dark Energy Survey where we are supplied with 898,963 51x51 grayscale images of either “fake” Supernova objects or supernovae of type 1A. I will be using 2 machine learning models, one being a Random Forest model’s output of 39 features from the autoscan project [4] introduced by Danny Goldstein from the University of California at Berkeley, and the other model is a Convolutional Neural Network (CNN) model supplied by a team of researchers from Dessa [5]. With the supplied CNN, I used 1 input layer, 2 hidden layers, and an output layer that takes in 2601 values of pixels from the images and determines the class for the given image with a cutoff probability of ~0.41. It uses a predetermined value of 210, 119, 63, and 25 nodes respectively for the convolutional layers.



Question 1. What is the variable the machine learner is supposed to predict? How accurate is the labeling? What is the annotator agreement (measured)?

Given a dataset of 898,963 51x51 images of potential Supernovae, the machine learner will attempt to classify whether the image is a Supernova of type 1A or not.

Since these images have gone through a thorough inspection via tens of professional Astronomers in a prominent astronomical survey project, each working on these classifications for more than a week, these images are very likely to be labelled correctly. However, given the difficult nature of classifying images based on whether they are supernovae or not, we will go ahead and assume that the labeling is ~96-98% accurate, an approximate accuracy value within the Astronomy field.

Lastly, due to the strict definition of rules to certainly verify an object to be a Supernova, the annotators majorly have the same agreements and will be assumed to be the same for our report.

Question 2. What is the required accuracy metric for success? How much data do we have to train the prediction of the variable? Are the classes balanced? How many modalities could be exploited in the data? Is there temporal information? How much noise are we expecting? Do we expect bias?

Due to the importance of classification of Supernovae within the field of Astrophysics, the required accuracy metrics for success are the classification accuracy, its respective generalization value for an estimate of memorization (given that the generalization value of $G > 1$, which we are attempting to have since it will display the model learning rather than memorizing the input data), false positive rate, along with its recall and precision values. Primarily, a successful classification rate of 95% and above is considered to be successful due to the similar accuracy values gotten from the other studies done on the dataset. I will be using a 90/10 percentage of the data for training (~805,000 images) and validating (~85,000 images) the predictions of the variables, respectively. Moreover, with a split rate of 50.51% and 49.49% for the distribution of the Supernovae and fake Supernovae images, respectively, the classes are pretty well balanced and should not significantly harm our results. Looking at the output of the Random Forest dataset that contains 39 distinct engineered features, a particular modality is present within the MIN_DISTANCE_TO_EDGE_IN_NEW field, a value describing the minimum distance from the the object to the edge, a value that can be commonly exploited since supernovae are brighter in images and the effects of the sun rays decrease the values, which interestingly enough seems to be almost 50% with the value of 0, whilst as non-zero values occupy the other half. In addition to that modality, there a few less

important modalities present within the dataset, which are the N3SIG3SHIFT, N2SIG5SHIFT, and N2SIG3SHIFT variables (further described in the research paper [6]). Furthermore, although the database does certainly contain the temporal information for the given images, they are unfortunately not given within the dataset and hence, are not able to be used. Since images are very common to be noisy when taking images during foggy nights, the images span a signal-to-noise ratio all the way from 0.01 to 1.00, which are taken into account by the given dataset. Lastly, we should not be expecting any bias other than the measurement bias induced within the images, due to a large span of time between the earliest and latest image within the dataset.

Question 3. What is the Memory Equivalent Capacity for the data (as a dictionary). What is the expected Memory Equivalent Capacity for a neural network?

Using the first algorithm in Chapter 9 from Professor Friedland's book (as displayed below), I was able to calculate an MEC of 1380 bits for the dataset with 39 features as a dictionary. However, for the implemented Neural Network as described above, the expected MEC is 546627 bits for the dataset. I was able to get this MEC value by using Professor Friedland's mentioned theorems in his published article [7] (In summary, (51*51*210)+min(51*51*210, 210)+min(210,119)+min(119,63)+min(63,25) = 546627).

Algorithm 1 Calculate the Memory Equivalent Capacity needed for a binary classifier assuming weight equilibrium in a dot product.

Require: *data*: array of length n contains d-dimensional vectors x, *labels*: a column of 0 or 1 with length n
  function *memorize*((*data*, *labels*))
    $thresholds \leftarrow 0$
    for all *rows* **do**
      $table[row] \leftarrow (\sum x[i][d], label[row])$
      $sortedtable \leftarrow sort(table, key = column\ 0)$
      $class \leftarrow 0$
    end for
    for all *rows* **do**
      if not $sortedtable[row][1] == class$ **then**
        $class \leftarrow sortedtable[i][1]$
        $thresholds \leftarrow thresholds + 1$
      end if
    end for
    $mec \leftarrow \log_2(thresholds + 1)$
  end function: *mec*

Question 4. What is the expected generalization in bits/bit and as a consequence the average resilience in dB? Is that resilience enough for the task? How bad can adversarial examples be? Do we expect data drift?

Using the equations retrieved from Brainome's documentation glossary [8] attached below, we expect to have a generalization value of 1.64 bits/bit and as a consequence, an average resilience of -4.32 dB. Moreover, please note that Brainome predicted an ideal expected generalization of 402.18 bits/bit and an average resilience of -2.60 dB. The resilience is unfortunately not enough due to the noisy nature of a significant fraction of the images within the dataset, hence some predictions will be impacted by them. As mentioned repetitively so far, a primary adversarial example will certainly be a noisy image due to its significance, hence affecting the predictions of the CNN model more than the Random Forest model since the CNN model uses the image data whilst as the Random Forest model uses primarily the metadata of the images. Since the basis of these values have remained unchanged for the past decade of the Astronomical survey, we should not expect any form of data drift other than the presence of noise due to advancements in Astronomical imagery.

$$Generalization = \frac{\#\ correctly\ classified\ instances}{MEC} \qquad Resilience = 20\ log_{10}\left(\frac{MEC}{\#\ correctly\ classified\ instances}\right)$$

Question 5. Is there enough data? How does the capacity progression look like?

After having fed the optimal 39 features from the Random Forest Model to Brainome, I was able to retrieve the capacity progression values and ultimately deduce that we have enough data, as implied by the convergence of the progression plot displayed below. This helps us notice that any more data than 80% of our input set is essentially not helpful to our classification model, since the model has already found the rule needed for the dataset. Lastly, using the second algorithm mentioned in Chapter 9 of Professor Friedland's book [9] displayed below, along with the help of Eleanor's Piazza post [10], I estimated the MEC correlating with the given percentage of the training data used in the plot displayed below.

Algorithm 2 Calculating the capacity progression for the Equilibrium Machine Learner.

Require: *data*: array of length n contains d-dimensional vectors x, *labels*: a column of 0 or 1 with length n

Require: $getSample(p)$ returns p percent of the *data* with corresponding *labels*.

Require: $memorize(data)$, see Algorithm 1.

procedure $CapProg((data, labels))$

$sizes = \{5, 10, 20, 40, 80, 100\}$

for all *sizes* **do**

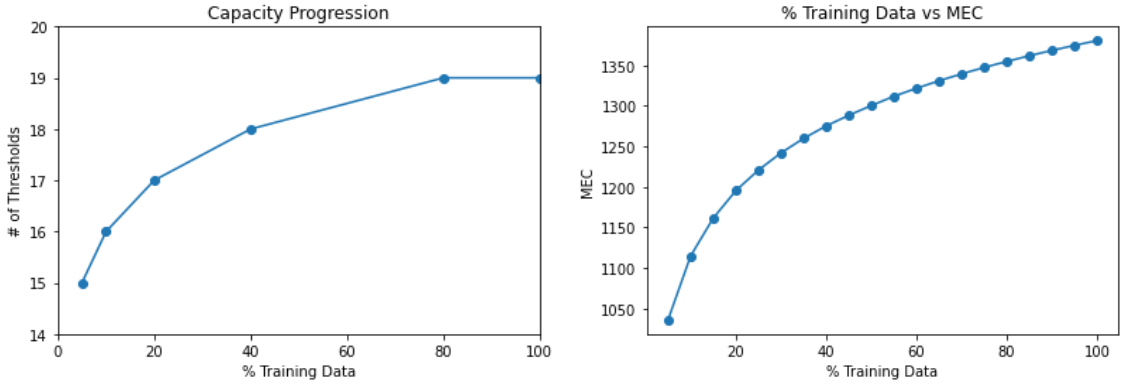
$subset = getSample(size)$

$mec = memorize(subset)$

print$"MEC\ for" + size + "\%\ of\ the\ data:\ " + mec + "bits"$

end for

end procedure



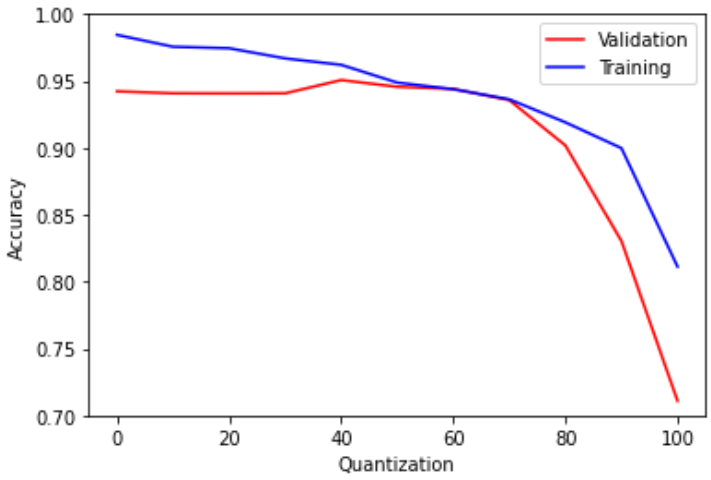
Question 6. Train your machine learner for accuracy at memory equivalent capacity. Can you reach near 100% memorization? If not, why (diagnose)?

Using the *-nosplit* flag mentioned in the Brainome documentation page, a flag that essentially prevents brainome from splitting the training data to automatically create a validation set, I was able to train the machine learner to reach a 99.9% accuracy that indicates that our model has reached near 100% memorization on the CSV of 39 features file. Moreover, using the pre-trained CNN model defined earlier, I was able to achieve ~96.5% accuracy by training the data for 40 epochs, which is also accurate and near enough to reach 100% memorization.

Question 7. Train your machine learner for generalization: Plot the accuracy/ capacity curve. What is the expected accuracy and generalization ratio at the point you decided to stop? Do you need to try a different machine learner? How

well did your generalization prediction hold on the independent test data? Explain results. How confident are you in the results?

I went ahead and trained my machine for generalization as displayed below, where using the N_approx variable mentioned in lectures and in the following Piazza post [11], the expected accuracy is ~95% and generalization ratio is 248.99 bits/bit. Using these values, it seems like the model is able to generalize very well and use what it has learned to correctly classify images, since the generalization ratio G is much greater than 1, a value that we wanted to be above. Moreover, it seems like this generalization value is also pretty well set for the last testing batch that we use, since the accuracy of the model on the testing batch is approximately 93.5% on average, which is a bit less than the training accuracy, along with the fact that an MEC of 889 bits is better than what was needed from above (~1400 bits). However, since the validation and training curves don't seem to correlate as well as I'd hoped it would (especially with the one from Piazza), I am not too confident in the results and would like to try these out with pre-defined CNN models for further justification and correction.



Question 8. Comment on any other quality assurance measures possible to take/ the authors should have taken. Are there application-specific ones? If time is present: How did you deal with it?

There are certainly a lot of other possible quality assurance measures that I could've taken throughout the process. Moreover, I could analyze more of the pre-defined and well-known machine learners, mainly within the ResNet series, to see if I could possibly define a more robust system that could challenge the Random Forest model. Something that might be even more surprising and most helpful to know is that there are other types of Supernovae (Type 1B, type 2, etc.) which were not considered within this dataset. Although the given dataset does not contain these types, I could have a look at the Zwicky Transient Facility's [12] dataset via Caltech's dataset API to either test my

model against other types of supernovae, or attempt to make a multiclass classification model. This would be particularly helpful with the fact that other types of supernovae are hard to differentiate, other than having a further analysis within their light curves and color filter information, something that was not thoroughly analyzed in this report. Regardless, we have built a strong binary classifier that can analyze a slightly-noisy image and cancel out the noise to very accurately classify the image and save a significant time from Astronomers' times.

Question 9. How does your experimental design ensure repeatability and reproducibility?

Along with the fact that this report has explained the methods used to reach to certain conclusions, the code used for defining the mentioned algorithms within the report, producing all the plots, and creating the CNN model is located in a repository on GitHub [13] for reproduction. Due to its primarily deterministic behavior, the supplied code, the pretrained features CSV files, and Brainome platform can be repeated for testing purposes. Lastly, the documentation used within this paper has been mentioned in the References section of the report for any further inspection if need be.

References:

1. "Autoscan." *Autoscan - Random Forest-Powered Artifact Rejection*, https://portal.nersc.gov/project/dessn/autoscan/
2. Dessa. "How 3 Engineers Built a Record-Breaking Supernova Identification System with Deep Learning." *Medium*, Dessa News, 27 May 2021, https://medium.com/dessa-news/space-2-vec-fd900f5566
3. "Home." *The Dark Energy Survey*, 14 Mar. 2017, https://www.darkenergysurvey.org/
4. "Autoscan." *Autoscan - Random Forest-Powered Artifact Rejection*, https://portal.nersc.gov/project/dessn/autoscan/
5. Dessa. "How 3 Engineers Built a Record-Breaking Supernova Identification System with Deep Learning." *Medium*, Dessa News, 27 May 2021, https://medium.com/dessa-news/space-2-vec-fd900f5566
6. Page 12, Goldstein, D. A., et al. "Automated Transient Identification in the Dark Energy Survey." *ArXiv.org*, 21 Dec. 2015, https://arxiv.org/abs/1504.02936
7. Friedland, Gerald, et al. "A Practical Approach to Sizing Neural Networks." *ArXiv.org*, 4 Oct. 2018, https://arxiv.org/abs/1810.02328
8. "Documentation - Access All BRAINOME Resources and Documentation." *Brainome.ai*, 23 Sept. 2021, https://www.brainome.ai/documentation/

9. Friedland, Gerald, et al. “An Information View on Data Science”, *Piazza.com*, Chapter 9, https://cdn-uploads.piazza.com/paste/jeqdgp7ec8fv4/bb8d09923f28f8df1326688d66f267909167aa329fce3b0f33616c54e562338e/Information_View_on_Data_Science.pdf
10. Piazza.com, https://piazza.com/class/kss3z4d4x027ag?cid=90
11. Piazza.com, https://piazza.com/class/kss3z4d4x027ag?cid=90
12. “Zwicky Transient Facility.” *IRSA*, https://irsa.ipac.caltech.edu/Missions/ztf.html
13. Abelo9996. “ABELO9996/CS294-082-Project: Repository for the Final Project of the CS294-082 Course at UC Berkeley.” *GitHub*, https://github.com/Abelo9996/CS294-082-Project