Addressing Safety Critical ML Applications Through Learned Skepticism

Tommy Galletta, Sammy Boddepalli, Anwuli Ajabor

Introduction

Through the research and experiments we plan to conduct during this project, our group plans to tackle the problem of applying machine learning to safety critical scenarios. This problem is one that has been long-standing in the machine learning community, as the requirements for error rate in safety critical systems can sometimes be as low as one out of every one billion uses of the software, whereas most machine learning systems usually struggle to achieve error rates even as low as one in one thousand. Our approach to this problem involved augmenting ML applications to have a level of skepticism in its predictions.

This research is important because it marks another step toward a solution to the problem of how ML applications can be utilized in a safety critical environment. In an ideal scenario, an ML system would be able to handle tasks where the result is “easy” to come to, while being able to recognize when a particular set of data is “too hard” to make a confident prediction on, in which case the data could be set aside for human review. This would allow for the automation that is afforded by ML systems to be applied to safety critical scenarios, while minimizing the risk of errors arising.

We plan to address this problem by experimenting with techniques where we enable a ML system to be self-skeptical. This could mean a variety of things, from allowing an ML system to refuse to make a prediction if it is skeptical of the prediction it would make, enabling the model to provide a confidence score alongside its prediction for the sake of user trust and transparency, some combination of both methods, or perhaps something else entirely. By the end of the project, we hope to compare how different approaches to self-skepticism impact the performance of the model, while taking into account both the change in accuracy, as well as the rate in which the model will consider its prediction to be of low confidence.

Literature Survey

The integration of machine learning systems into safety-critical applications necessitates a fundamental reconsideration of traditional deployment paradigms. While contemporary machine learning models have demonstrated remarkable performance across numerous domains, their application in high-stakes environments where error rates must approach one in one billion operations remains constrained by inherent limitations in reliability and predictability. The concept of prediction systems that have the ability to reject a prediction is far from new. The idea was discussed, and the concept formalized mathematically, over half a century ago [1].

Building upon these classical foundations, Qiao and Valiant addressed selective prediction in sequential contexts where algorithms observe data streams and maintain autonomy over both the timing and scope of predictions [2]. Their work represents a significant theoretical advancement by demonstrating that meaningful accuracy can be achieved for broad classes of statistical measures even without distributional assumptions regarding input data.

Gelbart and El-Yaniv [3] further enriched the theoretical landscape by establishing fundamental connections between agnostic selective classification, active learning methodologies, and the disagreement coefficient using pointwise-competitive classifiers which reproduce the decisions of the optimal classifier whenever they choose not to reject, with performance quantified through the probability measure of rejected instances.

Translating theoretical principles into practical implementations, SelectiveNet was developed: A neural network which leveraged an integrated option to refuse to predict on a given sample into its architecture [4]. This paper was able to show that by allowing for some amount of allowed rejections, specified by a “target coverage” hyperparameter, that a model could achieve much more reliable results in the samples that it provided predictions for. The group was able to apply the method to both classification and regression models, and in both cases more accurate results were observed when the model was allowed to make a small percentage of rejections. In the case of SelectiveNet, the greatest improvement with the least impact from diminishing returns seemed to occur at around a 15% target rejection rate.

Expanding the scope of rejection-capable systems beyond classification, Asif and Minhas [5] introduced a unified neural framework applicable to both classification and regression tasks.This dual framework of predictor model and rejector model enables machine learning systems to emulate human expert behavior by abstaining from low-confidence predictions, with the generalized formulation extending rejection capabilities beyond classification to encompass regression, thereby providing a unified treatment of uncertainty-aware prediction across multiple task domains.

However the reliability of confidence-based rejection mechanisms depends critically on model calibration. The paper by Guo et al. [6] highlights this by explaining how more often there

is not, there tends to be a discrepancy between model confidence and model accuracy. Guo et al. dubbed models where this is the case to be “miscalibrated”. They explain that a model whose prediction confidence closely matches its accuracy helps the user understand how likely it is that a given prediction is correct or incorrect. This is clear when you consider that, ideally, a model prediction of 50% confidence should be correct 50% of the time, and this logic should apply to any confidence value. The paper goes on to explain a variety of approaches to address calibrating the model, ultimately landing on temperature calibration to be the method that is the most reliable, computationally efficient, and easiest to implement. This calibration method simply involves multiplying the values leading into the final softmax layer of a classification model by some constant 1/T, where T is the “temperature” of the model. The model is trained with a temperature of 1, and then this temperature value can be adjusted in testing such that the model’s confidence more closely matches its accuracy at that confidence.

And concerning other rejection mechanisms Hendrickx et al. [7] formalized paradigms for ambiguity rejection, which occurs when training assumptions regarding sampling distributions or class balance are violated, and novelty rejection, which addresses instances from previously unseen distributions encountered post-deployment.

A comprehensive overview of uncertainty in ML/DL is given by Fakour et al. [8], who classify it into distributional, aleatoric, and epistemic forms and examine a variety of quantification techniques, including ensembles, Bayesian inference, and conformal prediction. Sluijterman et al. [9] address the problem of assessing uncertainty by criticizing traditional metrics like log-likelihood and marginal coverage for regression tasks, claiming that they mask pointwise failures. They suggest simulation-based testing in conjunction with more precise interval-based metrics for a more thorough assessment.

To complement these viewpoints, Wang et al. [10] address uncertainty from the standpoint of safety assurance by presenting Neurify, a formal verification framework that improves model interpretability and robustness by reducing overestimation errors and effectively identifying safety property violations. Collectively, these contributions highlight the need for rigorous evaluation strategies, formal verification tools, and improved definitions and quantification techniques to advance uncertainty research and close the gap between theoretical uncertainty modeling and practical safety-critical deployment.

In conclusion the literature study emphasizes that when implementing machine learning in safety-critical situations, uncertainty and selective prediction are essential requirements rather than incidental factors. While research on calibration [6] and ambiguity/novelty rejection [7] highlights the difficulties in ensuring such mechanisms function effectively under real-world distributional shifts, foundational works on selective prediction and rejection mechanisms demonstrate that allowing models to abstain when confidence is low can significantly improve reliability. These observations drive our project's investigation of self-skeptical machine learning

systems, or models that can express their uncertainty through hybrid techniques, calibrated confidence scores, or abstention from prediction.

Methodology

Data:

We evaluated our approach using the OCTMNIST dataset, a standardized subset of medical retinal Optical Coherence Tomography (OCT) images from the MedMNIST collection. The data consists of 109,309 28x28 grayscale image samples divided into four classes: normal retina, drusen, diabetic macular edema (DME), and choroidal neovascularization (CNV). To ensure compatibility with the ResNet-18 architecture, all images were preprocessed by converting from grayscale to three-channel RGB format. Standard tensor normalization was then used.

Baseline Model:

For this study, we employed a ResNet-18 architecture as our baseline model, a widely-adopted deep convolutional neural network known for its effectiveness in image classification tasks. ResNet-18, introduced by He et al. [11], consists of 18 layers organized into residual blocks that enable training of very deep networks through skip connections. The architecture comprises an initial convolutional layer, followed by four residual layer groups containing 2, 2, 2, and 2 residual blocks respectively, culminating in a global average pooling layer and a fully connected classification layer.

To adapt the standard ResNet-18 architecture to our classification task, we modified the final fully connected layer to output predictions for 4 classes rather than the original 1000 ImageNet classes.

The model was initialized without pre-trained weights (trained from scratch), and all parameters across all layers were set to be trainable, allowing the network to learn task-specific feature representations throughout the entire architecture. The model was deployed on google colab TPU for efficient training.

The model was trained for 100 epochs using the Adam optimizer with an initial learning rate of 0.001. The loss function employed was categorical cross-entropy loss (CrossEntropyLoss), the standard choice for multi-class classification tasks, which computes the negative log-likelihood of the correct class given the model's softmax predictions. To improve convergence and prevent overfitting in later training stages, we implemented a learning rate scheduler that decays after 50 and 75 epochs by 0.1.

Calibration of baseline model:

The severe overfitting observed in the baseline model has direct implications for calibration performance. The overfitting problem compounds the calibration issue as not only is the model making incorrect predictions on test data, but it is also extremely confident in these incorrect predictions, resulting in both poor accuracy and poor calibration.

To address this calibration problem, we employed temperature scaling, a simple yet highly effective post-processing calibration technique proposed by Guo et al. [6]. For classification problems, the neural network output a vector known as the logits. The logits vector is passed through a softmax function to get class probabilities. Temperature scaling simply divides the logits vector by a learned scalar parameter, i.e. We learn this parameter on a validation set, where T is chosen to minimize negative log likelihood. Intuitively, temperature scaling simply softens the neural network outputs. This makes the network slightly less confident, which makes the confidence scores reflect true probabilities. In our case the temperature parameter T was optimized using the validation set to minimize the negative log-likelihood (NLL) via the L-BFGS optimizer with a maximum of 200 iterations.

Implementation of SelectiveNet:

In order to solve the related issues of prediction uncertainty quantification and selective prediction, our solution expands upon the SelectiveNet architecture put forward by Geifman and El-Yaniv [4]. This design, CalibratedSelectiveNet, has three essential elements that work together to facilitate self-skeptical prediction. The first part uses a pretrained ResNet-18 model as a feature extraction foundation. The second is a calibrated prediction head that applies temperature scaling to class predictions. During this step, the raw neural network logits are converted to calibrated confidence scores using the temperature scaling technique found in Guo [6], resulting in the model outputting confidence scores that more accurately reflect the accuracy of the prediction. Following initial model training, the temperature parameter is tuned against the validation set to reduce the discrepancy between the model's prediction confidence and its accuracy.

The third element is a trained selection mechanism, modeled after that in Geifman and El-Yaniv [4], made up of a two-layer neural network with batch normalization, sigmoid output, and ReLU activation. For every input, this selection head outputs a selection score ranging from zero to one, which represents the model's confidence in generating a prediction as opposed to abstaining. This selection mechanism is placed as a secondary branch after the feature representation layer of the ResNet-18 backbone. This secondary branch is then trained while all other layers in the model are frozen, such that the feature representations learned during initial training are preserved, and the selection mechanism learns to select based on these features.

To guarantee steady convergence and peak performance, training was done in two stages. In the initial stage, 72.2% test accuracy was attained by training the ResNet-18 backbone for standard classification on OCTMNIST with cross-entropy loss until convergence. Selective prediction learning was based on this pretrained model. Using a composite loss function that strikes a balance between prediction accuracy and selective coverage, we trained the CalibratedSelectiveNet for 75 epochs in the second phase, building on the pretrained backbone. We calibrated the temperature scaling on the validation set after training. A limited-memory optimization approach with a maximum of 100 iterations was used to improve the temperature parameter to reduce cross-entropy loss. Expected Calibration Error (ECE), which divides predictions into 15 confidence bins and calculates the weighted average of the difference between accuracy and confidence within each bin, is how we assessed calibration quality. With lower values denoting greater calibration, this statistic measures how closely the model's stated confidence matches its actual prediction accuracy.

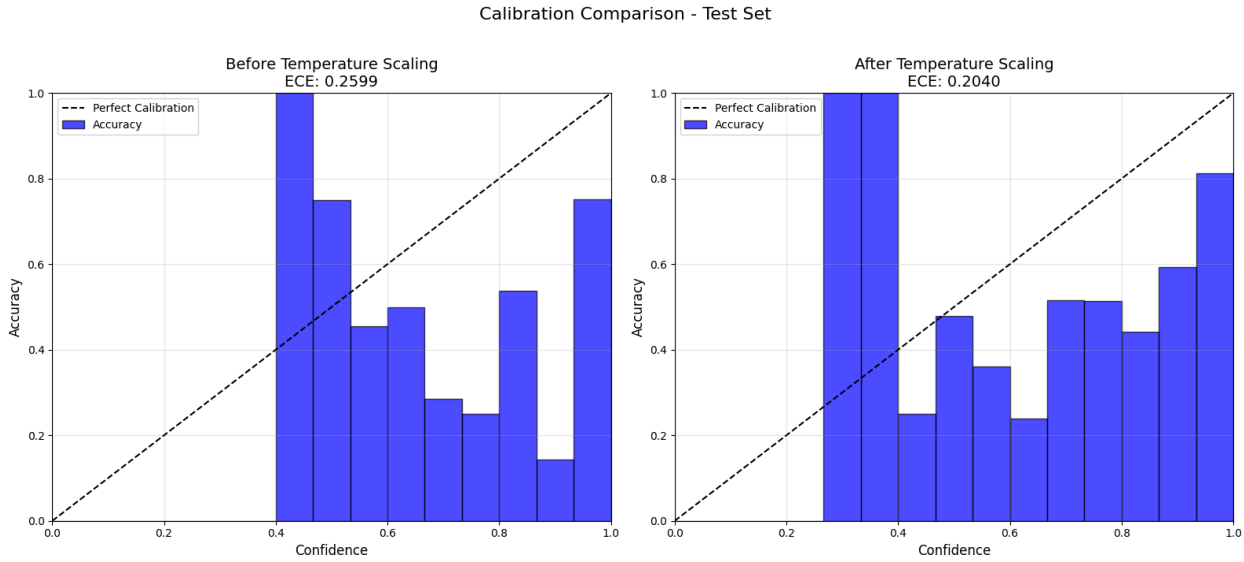
To thoroughly test the efficiency of our self-skeptical method, model performance was evaluated in a number of dimensions. We examined coverage, which is the percentage of samples accepted, rejection rate which is the percentage of samples where the model refrains from making a prediction, and selective accuracy, which is the accuracy of the model, considering calculated samples where the model decides to make a prediction.

Results and Discussion

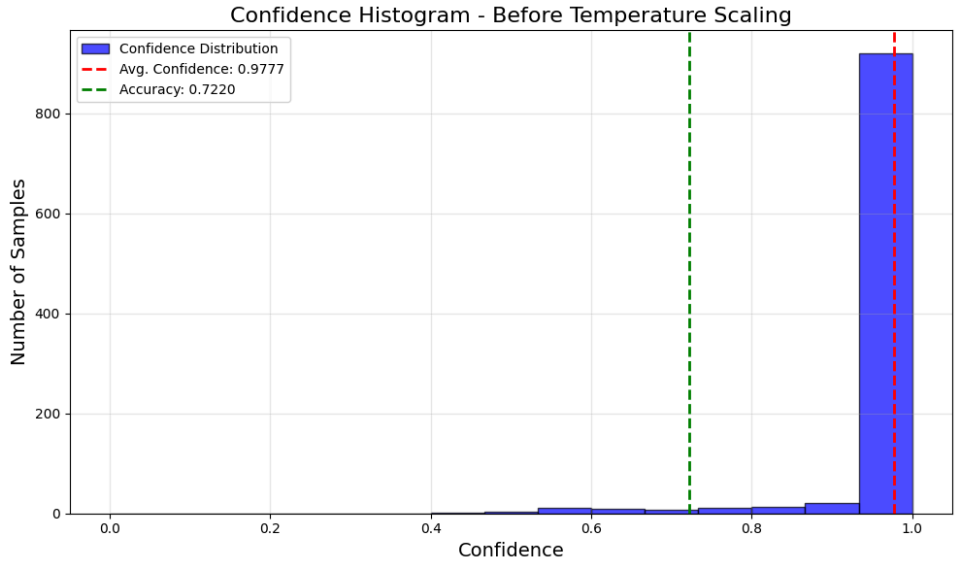
Upon evaluation on the held-out test set, the baseline model achieved an accuracy of 72.20% with a negative log-likelihood (NLL) of 3.5095. This represents a dramatic performance drop of 27.56 percentage points from training accuracy (99.76%) and from validation accuracy (94.21%), revealing severe overfitting to the training data. The test set NLL of 3.5095 and the subsequent calibration analysis (ECE = 25.99%) indicate that the model not only fails to generalize accurately but also produces highly miscalibrated confidence estimates. The near-perfect training accuracy (99.76%) likely resulted in the model learning to output very high confidence scores during training, which do not translate to test examples where accuracy drops to 72.20%. This disconnect between confidence and accuracy manifests as the severe overconfidence observed in the reliability diagrams, where the model predicts with >95% confidence while achieving only 72-75% accuracy on many test samples .

The application of temperature scaling to the test set yielded significant improvements in calibration metrics while maintaining classification performance. As summarized in Table 1, the model's accuracy remained unchanged at 72.20%, confirming that temperature scaling preserves the model's discriminative ability. However, the ECE decreased from 25.99% to 20.40%, representing a 21.5% relative improvement (5.59 percentage points absolute reduction). The

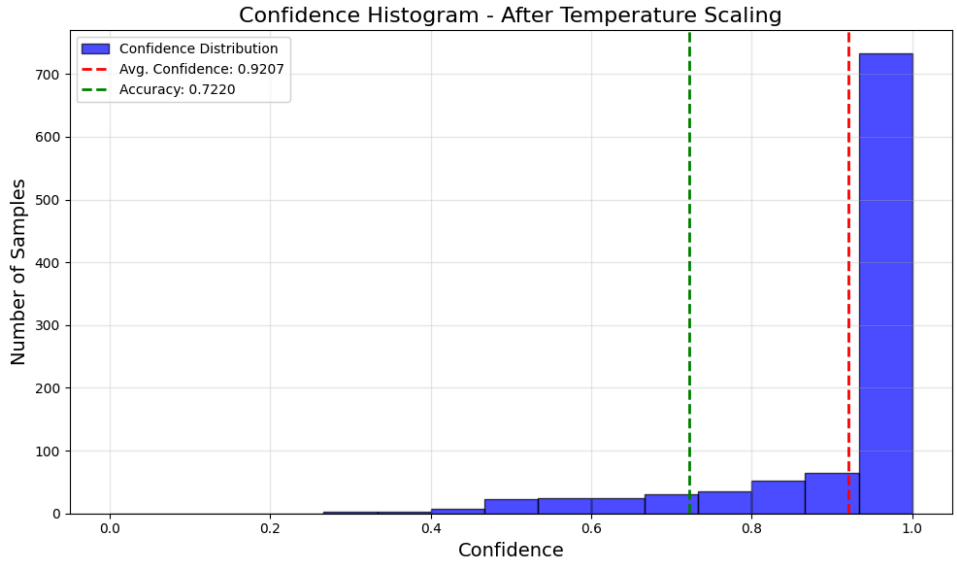
NLL showed dramatic improvement, decreasing from 3.5095 to 1.1546, a reduction of 2.3549, indicating substantially better-calibrated probability estimates.



In Figure 1, the left panel reveals the uncalibrated model's significant deviation from perfect calibration (dashed diagonal line), with blue bars representing actual accuracy falling consistently below expected confidence levels. Whereas the right panel demonstrates the effect of temperature scaling while not achieving perfect calibration (ECE = 20.40%), the accuracy bars more closely align with the perfect calibration line, particularly in the mid-to-high confidence range (0.4-1.0). The reduction in the gap between observed and expected performance indicates improved reliability of the model's confidence estimates.



In Figure 2, the baseline model exhibited extreme overconfidence, ranging from 0.4-1.0 with the vast majority of predictions concentrated at very high confidence levels (>95%), as shown by the large spike near 1.0. The average confidence of 97.77% drastically exceeded the actual accuracy of 72.20%, creating a confidence-accuracy gap which is a clear indicator of severe miscalibration.



After temperature scaling (Figure 3), the confidence distribution became substantially more spread and realistic. While the highest concentration still occurred at high confidence levels, the distribution broadened significantly across the 0.2-1.0 range, and the average confidence decreased to 92.07%, reducing the confidence-accuracy gap .

We successfully proved our hypothesis that using calibration in conjunction with accept/reject techniques would result in notable increases in accuracy while preserving respectably low rejection rates. In contrast to the 72.2% baseline with no rejections, we obtained 77.92% accuracy with 21.2% rejects using a CalibratedSelectiveNet architecture with temperature scaling and a learnt selection head on ResNet-18 trained on OCTMNIST data. an improvement of 5.72% percentage points. This can also be thought of from the perspective of reduction of error, where originally 17.8% of predictions were erroneous with the baseline model, only 11.08% of predictions were incorrect after calibration and selection, meaning approximately 37% of the previously incorrect predictions were able to be correctly handled by the revised model, at the cost of a 21.2% rejection rate.

This shows that calibrated selective prediction offers a workable strategy for implementing self-skeptical machine learning systems in safety-critical medical applications,

effectively striking a balance between increased accuracy and realistic coverage requirements.

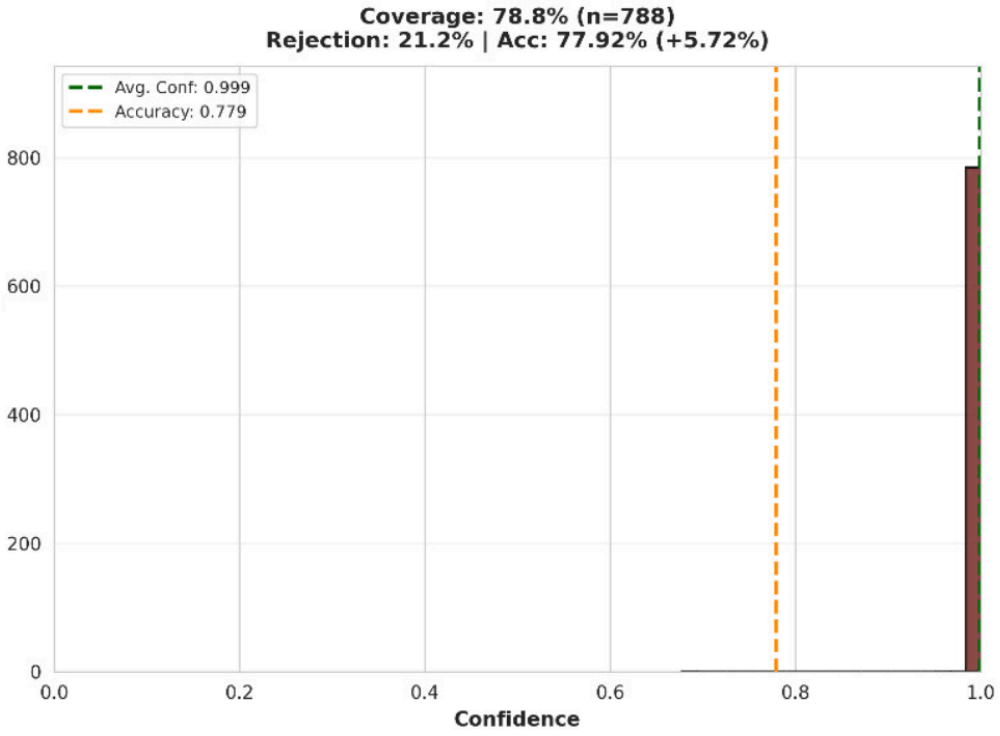


Figure 4, Result of applying selectivenet

Future Work

Conclusion

References:

[1] C. Chow, “On optimum recognition error and reject tradeoff,” IEEE Transactions on Information Theory, vol. 16, no. 1, pp. 41–46, Jan. 1970, doi: https://doi.org/10.1109/tit.1970.1054406.

[2] M. Qiao and G. Valiant, “A Theory of Selective Prediction,” arXiv.org, 2019. https://arxiv.org/abs/1902.04256

[3] Roei Gelbhart and Ran El-Yaniv, “The Relationship Between Agnostic Selective Classification, Active Learning and the Disagreement Coefficient,” Journal of Machine Learning Research, vol. 20, no. 33, pp. 1–38, 2019, Accessed: Oct. 01, 2025. [Online]. Available: https://jmlr.org/papers/v20/17-147.html

[4]Y. Geifman and R. El-Yaniv, “SelectiveNet: A Deep Neural Network with an Integrated Reject Option,” arXiv.org, 2019. https://arxiv.org/abs/1901.09192

[5] A. Asif and Minhas, “Generalized Learning with Rejection for Classification and Regression Problems,” arXiv.org, 2019. https://arxiv.org/abs/1911.00896

[6] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On Calibration of Modern Neural Networks,” arXiv:1706.04599 [cs], Aug. 2017, Available: https://arxiv.org/abs/1706.04599

[7] K. Hendrickx, L. Perini, Van, W. Meert, and J. Davis, “Machine Learning with a Reject Option: A survey,” arXiv.org, 2021. https://arxiv.org/abs/2107.11277

[8] F. Fakour, A. Mosleh, and R. Ramezani, “A Structured Review of Literature on Uncertainty in Machine Learning & Deep Learning,” arXiv.org, 2024. https://arxiv.org/abs/2406.00332

[9] Laurens Sluijterman, E. Cator, and T. Heskes, “How to evaluate uncertainty estimates in machine learning for regression?,” Neural Networks, vol. 173, pp. 106203–106203, Feb. 2024, doi: https://doi.org/10.1016/j.neunet.2024.106203

[10] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, “Efficient Formal Safety Analysis of Neural Networks,” arXiv.org, 2018. https://arxiv.org/abs/1809.08098

[11] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Work Split:

Name	Task	Percentage
Tommy Galletta	Literature Survey	30%
	Introduction	90%
	Literature Write-up	20%
Sammy Boddepalli	Literature Survey	40%
	Introduction	5%
	Literature Write-up	40%
Anwuli Ajabor	Literature Survey	30%
	Introduction	5%
	Literature Write-up	40%