Adversarial attacks and defenses on neural network whale classifiers

Jerry Kurtin
Georgetown High School
Georgetown, Texas

Jeriah Yu
Liberal Arts and Science Academy
Austin, Texas

Supervisors
Reid Wyde, Scott Johnston
Signal and Information Sciences Laboratory

This page intentionally left blank.

Adversarial Attacks Applied to Whale-Detecting Neural Network

TEXAS
The University of Texas at Austin

Jerry Kurtin – Georgetown High School – Georgetown, TX
Supervisors: Reid Wyde and Scott Johnston
Signal and Information Sciences Laboratory

Background

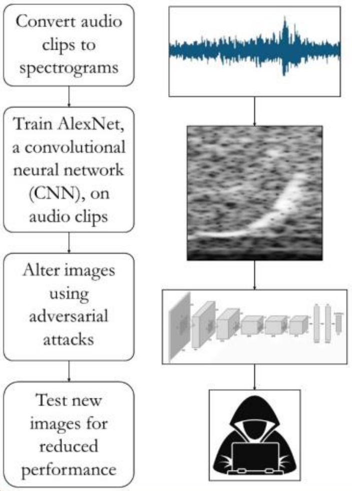
- Modern neural networks tend to be susceptible to adversarial attacks.
- Adversarial attack: a small, targeted disruption to an input image that causes a model to misclassify the image
- Adversarial attacks could cause real-world damage as important technology begins to rely on machine learning.

Dataset: 30,000 2-second audio clips from ocean buoys run by Cornell University

Objective

- Create a neural network that can distinguish North Atlantic right whale calls from ocean noise and other whale calls
- Discover vulnerabilities in the model through white-box and black-box attacks

Process



Convert audio clips to spectrograms

Train AlexNet, a convolutional neural network (CNN), on audio clips

Alter images using adversarial attacks

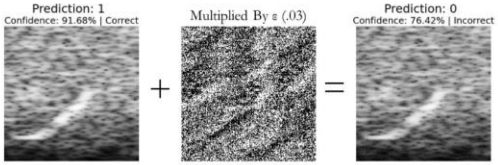
Test new images for reduced performance

Attack | Original Image | Perturbation | Perturbed Image

White Box Attacks: unrestricted access to model

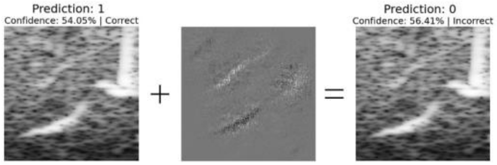
Fast Gradient Sign Method (FGSM)

- Calculates model gradient on image
- Steps in the opposite direction by a constant ε
- Computationally cheap, but noticeable



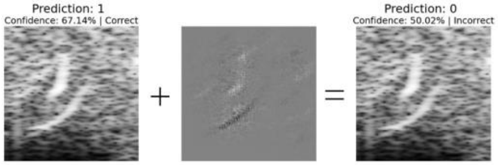
Carlini and Wagner (CW)

- Searches for smallest change to image that causes misclassification
- 1000 steps taken
- Computationally expensive, but more effective and less noticeable



DeepFool

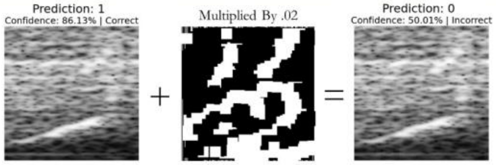
- Finds nearest hyperplane
- Calculates the changes needed to cross the hyperplane
- Hyperplane: a high-dimensional 'line' separating different classifications
- Efficient and subtle



Black Box Attacks: Only given access to a model's final decision and certainty

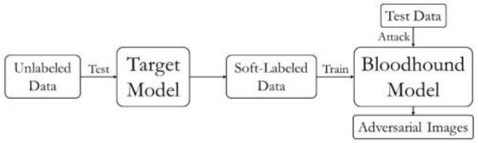
Square

- Changes a random square of pixels
- Tests for reduced certainty in model
- Repeats until successful misclassification
- "Guess and check"

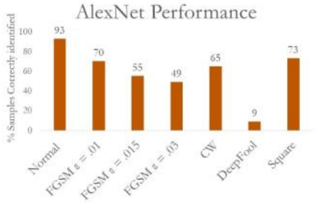


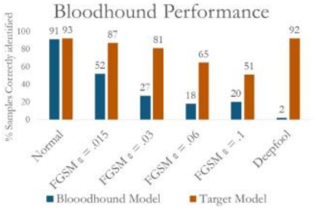
Bloodhound

- Labels spectrograms with output of target model
- Trains a 'bloodhound' model on labeled outputs
- Performs white-box attacks on bloodhound model



Results





Conclusion

- Image-recognition CNNs can be accurately used for sound classification
- White and black box attacks succeeded in reducing accuracy below random chance
- Decision borders are cloudy due to small dataset

Moving Forward

- Bootstrap dataset to train generalization
- Create realistic attacks that perturb original sound samples
- Expand network to detect and identify animal calls and human activity

Acknowledgements

I'd like to thank Reid Wyde, Scott Johnston, Anna Chaney, and Hector Gonzalez for their generous mentorship and patience.

This page intentionally left blank.

Adversarial Defenses on Neural Network Marine Mammal Classifiers

Jeriah Yu - Liberal Arts and Science Academy, Austin, Texas
Supervisors: Reid Wyde, Scott Johnston
Signal and Information Sciences Laboratory, Applied Research Laboratories

Objective

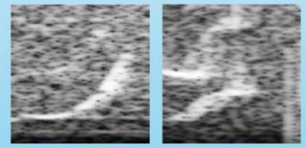
Investigate methods that increase robustness of neural networks against adversarial attacks in audio classification.

Purpose

- Identify underwater sound patterns
- Neural networks for high accuracy classification
- Training requires large amounts of labelled data
- Convolutional networks for images via spectrogram
- Deep networks susceptible to random and adversarial perturbations

Data

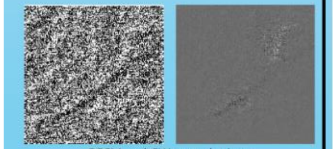
Cornell buoy audio set: 2-class (binary presence of North Atlantic Right Whale), uniform 1 sec, 2kHz SR
10 epochs, 80/20 split
Data weighting



Whale present vs not present

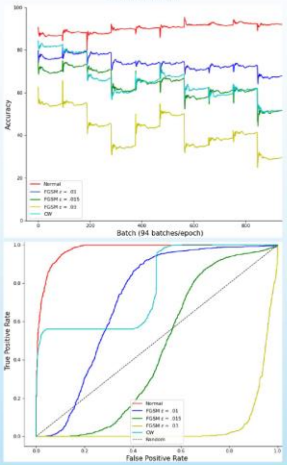
Attacks

FGSM, CW, DeepFool, Square, BH

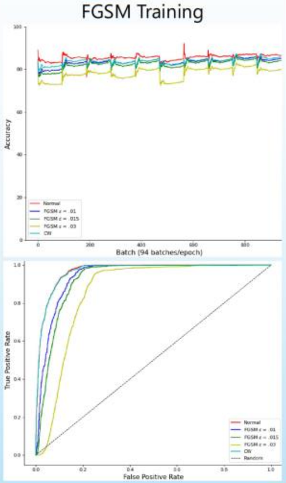


FGSM and CW perturbations

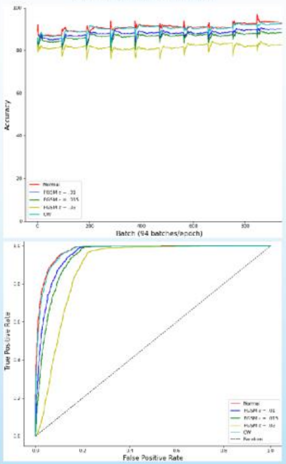
Baseline



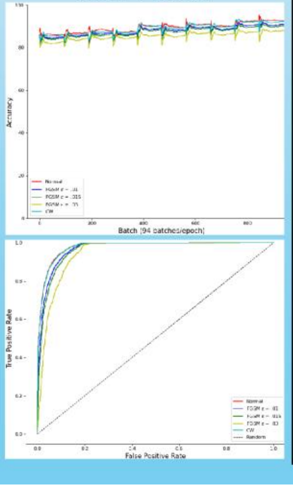
FGSM Training



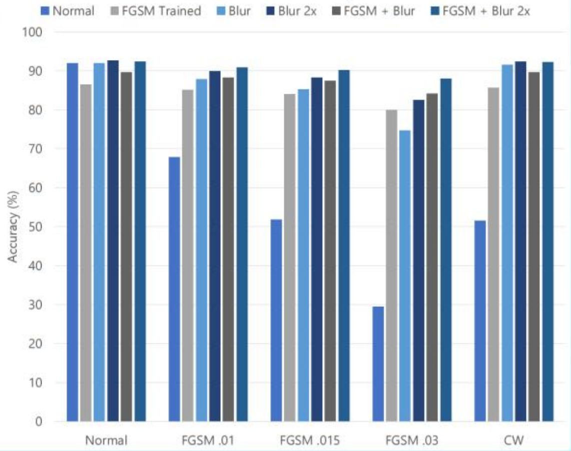
2X Gaussian Blur



FGSM + 2X Blur

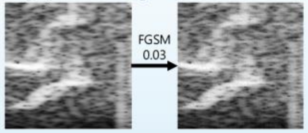


Defense Accuracy

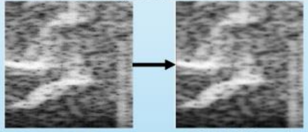


Defenses

Adversarial Training
Augment training data with perturbations on the fly
High compute cost, long training
Highly effective against attack



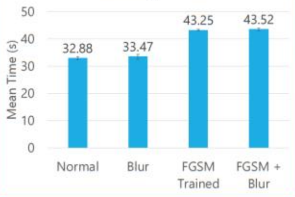
Gaussian Blur
Smoothens noise and perturbations
Dampens white-box attack gradient
Low computing cost, run on both sets



FGSM 0.03

FGSM 0.03 blurred

Training Time



Conclusion

- Image classification CNNs can be defended against attacks.
- Certain defenses are attack-specific.
- Defenses can combine to provide versatile coverage without sacrificing baseline model accuracy.
- Well-defended models can achieve normal accuracy against attacks.

This page intentionally left blank.

Adversarial attacks and defenses on neural network whale classifiers

Jerry Kurtin, Jeriah Yu
Georgetown High School, Georgetown, Texas
Liberal Arts and Science Academy, Austin, Texas

Supervisors: Reid Wyde and Scott Johnston, Signal and Information Sciences Laboratory

Surface and underwater ocean vessels have a need to quickly and accurately detect and identify sound patterns for species monitoring, obstacle avoidance, or identification of other vessels. This can be done by transforming audio into spectrogram images and processing them with convolutional neural network (CNN) models, taking advantage of recent advancements in image classification. While CNNs can achieve high classification performance over ideal data, recent work has shown that image classification models are susceptible to adversarial attacks, nearly-imperceptible image perturbations that significantly degrade model accuracy, precision, and recall. This project investigates the capabilities of a publicly available CNN, AlexNet, on a binary classification problem to detect whale presence from single-channel audio. The audio samples are Fourier transformed into a power spectral density array and up-sampled to fit the model dimensions of AlexNet, achieving ~93% accuracy. The trained models were then tested with perturbed images from multiple white-box attacks (with access to model output and model gradients during inference), and black-box attacks (with access to only model output). These attacks severely degraded the performance of the models. To combat this, various data preprocessing-based defenses were proposed and implemented, along with combinations of defenses. It was found that targeted, computationally intensive defenses improved accuracy significantly, while universal, cheaper defenses such as low-pass filtering could generalize better, defending against multiple types of attacks. Future work would involve expanding our models to classify multiple species of marine mammals and generating more realistic attacks on the raw audio.

I. Introduction

Underwater acoustical identification is a critical requirement in multiple fields such as species protection and population monitoring. This task requires large amounts of audio data to be analyzed and classified accurately, a difficult and time-consuming effort for human scientists. Instead, neural networks can be utilized to classify such audio data automatically to a high degree of accuracy without much human intervention.

In recent years, image classification neural networks have greatly improved. In order to take advantage of these existing optimized, high-performance network models, audio samples are converted into power spectral density (PSD) spectrograms which relate signal strength to frequency and time.

Spectrograms are then fed through convolutional neural networks (CNNs). A CNN is a deep neural network designed to effectively work on images. The first few layers convolve pixels together, reducing the size within each layer while retaining and extracting abstract features in two-dimensional layers before flattening to one dimension and determining a final confidence value.

Despite their strength, CNNs are vulnerable adversarial attacks: small, targeted, and nearly unpredictable disruptions to input images that cause the network to misclassify the input[1]. Adversarial attacks can be classified into two main styles: white-box attacks in which the attacker is granted full access to the model, and black-box attacks that only access a model's inputs and outputs. A third classification, grey-box, is a catch-all to describe attacks which require some additional information beyond a true black-box attack, such as the model's certainty of prediction. Figure 1 visualizes the spectrum onto which these attacks fall.[2]

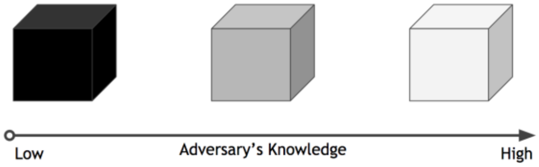


Figure 1: The gradient of attack types

These attacks disrupt neural networks from successfully learning and generalizing the task, but there are multiple proposed defenses against adversarial disruption. Two styles of defense are outlined in this paper: incorporating adversarial images in the training process and introducing a lossy transformation on all images entering the model in order to filter small perturbations.

II. Classifier

A. Dataset

The binary whale dataset that is used to train and test the model comes from the Cornell buoy North Atlantic Right Whale set.[3] This set consists of 30000 labelled samples and ~54000 testing samples without publicly released labels that were used for the Bloodhound black box attack.

The spectrograms used by the model are generated by converting the provided 2 second audio clips into a grayscale PSD spectrogram array using the Fast Fourier Transform. Decibel scaling was performed on the audio strength output values while linear frequency scaling was preserved due to apparent noise increase when converted to logarithmic scale.

The spectrogram values are z-score normalized to standardize across samples and saved as raw NumPy arrays of shape 100x55 rather than an image to preserve data without decreasing the signal-to-noise ratio. The data loader then upscales into a 224x224 "image" array to fit the dimension requirements of the models used for training and testing.

The training dataset contains a disproportionate number (~80%) of negative samples (no whale call present). This causes the model to naturally favor negative responses, missing a higher percentage of positive test samples than negative, reducing both precision and recall. To address this, each class was weighted $1/\sqrt{n}$, where n is the total number of species in the class. This causes the less prevalent positive samples to each have a stronger impact on the model parameters when backpropagated. The weighting slightly reduces the model's accuracy from 92.68% to 92.52%, increases recall, the proportion of true positives over all positive samples, from 80.16% to 87.40%, but reduced precision, the proportion of true positives over all positive predictions, from 87.03% to 81.44% due to a higher number of false positives. This brings the model's proportions of positive predictions closer to negative predictions (Fig. 2).

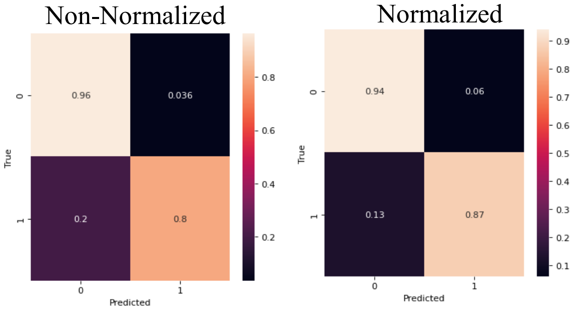


Figure 2: Confusion Matrices for Normalized Model

B. Network

AlexNet is a smaller image-classifying CNN designed in 2012.[4] Due to the small dataset and binary classes, newer and deeper networks such as ResNet,[5] MobileNet,[6] and VGG[7] were found to memorize and overfit the data, performing poorly in testing and remaining brittle to attacks even after defensive training.

AlexNet originally takes an input size of 224x224x3, a square RGB image, but we alter the architecture to fit our grayscale spectrograms (224x224x1). 5 convolution layers and 3 pooling steps reduce the size of each layer from 224x224 pixels (50,176 total) to a single layer of 4096 nodes. The complete architecture is visualized in Fig. 3.[8] By convolving the image instead of stringing out each individual pixel, the information about a pixel's relative location and the values of nearby pixels are preserved. This allows the network to view the input images more like a human would and learn larger patterns of the image instead of from each pixel independently.

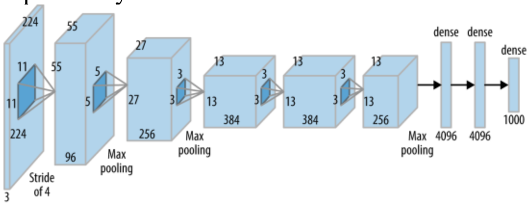


Figure 3: A visualization of the AlexNet Architecture

C. Addressing Data Leakage

Data leakage occurs when the training set and testing set for a model have overlapping data. In our first tests, we neglected to set a random seed when we partitioned our labeled dataset into training and testing sections (using the k-fold split technique with 3 folds). This caused the training script and testing script to have different random seeds and therefore select training and testing data groups independently, contributing to overlap.

To address this, a seed of 42 is set for all random elements in our project, and the answer to the ultimate question of life, the universe, and everything becomes the answer to our data leakage problems. As an additional measure, a random sample of 20% of the labeled data is permanently partitioned in a separate folder as the testing dataset, and the other 80% is used as the training dataset, similar to the hard partitions provided by the MNIST handwriting recognition dataset. Experimenting with different 20% samples of the data showed very little change between tests, so we assume that a random selection is a representative sample, and any similarities from different buoys/geographical locations do not affect our tests.

D. Initial Performance

Out of the 6,000 test samples, the AlexNet model classifies 92.52% of the samples correctly. The model's precision is 81.44%, and its recall is 87.40%. Our weighted model errs more towards positive predictions than the unweighted version, maximizing the number of true samples detected but classifying more samples as false positives. In practical applications like a passive monitoring system, whale calls would be an important rarity, so it would be worth sorting through a few false positives to catch all whale calls in the area.

Figure 4 top shows the testing accuracy of the baseline model increasing over the 10 epochs of training, reaching 92% at epoch 10. The bottom shows the receiver operating characteristic (ROC) curve which correlates false positive and true positive rates of the model. An ideal model would have 0.0 false positive and 1.0 true positive, while a random predictor would be represented by the black line, and anything below is worse than random. The area under the curve (AUC) from 0-1 represents the model's class distinguishing ability, with the baseline at 0.97.

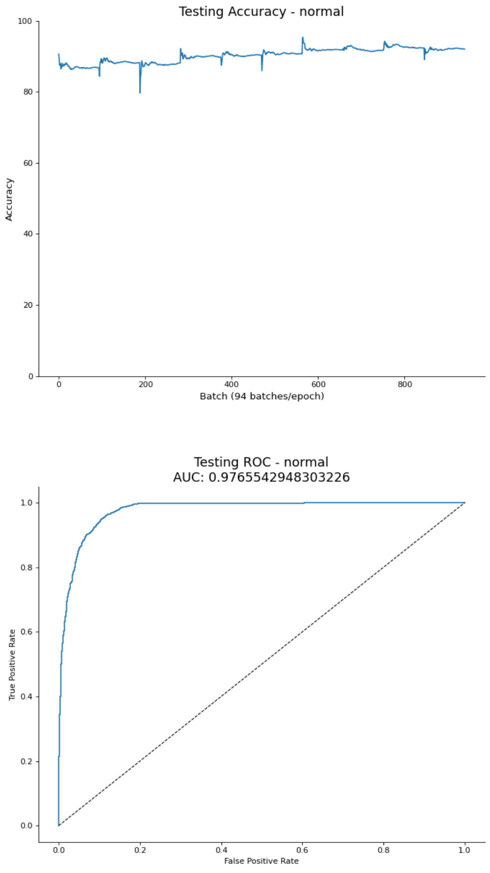


Figure 4: Baseline performance accuracy and ROC curve

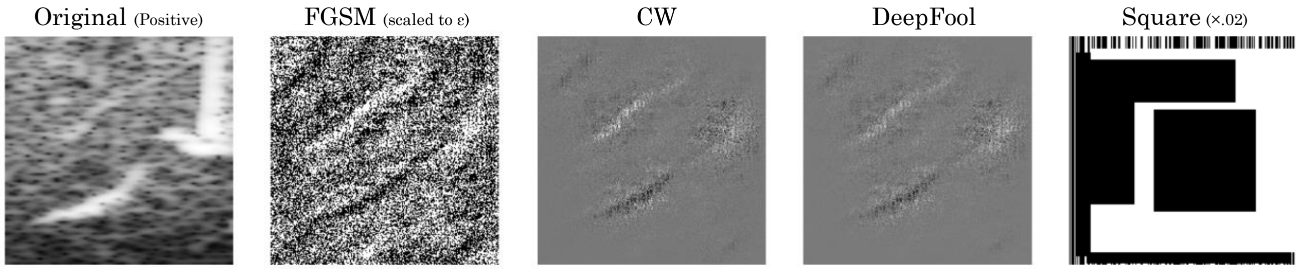


Figure 5: The perturbation added for misclassification in each attack

III. Attacks

With the model achieving high accuracy with standard training and testing, we perform multiple types of attacks against the classifier. Successful attacks prove the danger in relying on a classification model without defensive training (Fig. 6).

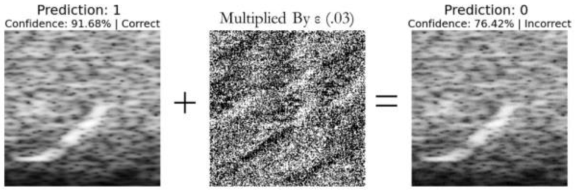


Figure 6: FGSM successfully confusing the model

A. White-Box

The simplest form of adversarial perturbation, white-box attacks require unrestricted access to a model's structure and gradients. While white-box attacks cannot practically attack a real-world model on their own, they provide important information on a model's robustness, and they can be fed into a model's training sequence to increase dataset size and reduce a model's brittleness to all attacks.

The first attack implemented is the Fast Gradient Sign Method.[9] In this attack, an image is passed through the target model to calculate the loss, and the loss is passed back the image, calculating the best change necessary in each pixel to correctly classify the image with the highest certainty. The attacker then moves the image in the opposite direction (gradient ascent), calculating the negative sign of this change and adding it (multiplied by a constant ε to determine the power of the attack), to the original image. While fast, this attack noticeably alters the image at ε > 0.015.

The attack proposed by Carlini and Wagner[10] provides a more sophisticated version of the FGSM attack. Referred to as the CW attack, it searches for the smallest possible perturbation that successfully misclassifies an image after a specified number of steps, 100 in our tests. The actual function, listed in Equation 1, is more complicated to increase efficiency, but performs the same process. This attack's perturbations are less noticeable to the human eye than FGSM (Fig. 5), but they take significantly more computation time (estimated 180x).

$$\text{minimize} \ \|\frac{1}{2}(\tanh(w)+1)-x\|_2^2 + c\cdot f(\frac{1}{2}(\tanh(w)+1)$$

with f defined as

$$f(x') = \max(\max\{Z(x')_i : i \neq t\} - Z(x')_t, -\kappa).$$

Equation 1: The CW Objective Function

Named DeepFool,[11] the final attack, utilizes a unique strategy. Instead of taking advantage of the model's gradient, it searches for the nearest hyperplane, the high-dimensional border that separates different classifications inside the model's probability space. Once it finds the closest hyperplane in Euclidian space, it calculates the orthogonal projection of the image's current values onto a point just beyond the hyperplane and adds the value of this vector to the original image. As a result, very little perturbation is required to misclassify the image (Fig. 5), but the attack only requires an estimated 9x more computation time than FGSM.

B. Black-Box

Unlike white-box attacks, black-box attacks can be realistically implemented on brand-name neural networks such as Google Lens or the computer vision networks in self-driving cars. Two such attacks are covered in this paper

Square[12] changes a small square of pixels uniformly on the input image then tests for reduced confidence, repeating this up to 7,500 times until misclassification is reached. This attack is technically grey-box, for it requires the confidence of a model's prediction, but this information is available for some public models. However, the large number of queries to the model required to find adversarial images drastically increases runtime to 1700x that of FGSM and would likely be detected as adversarial behavior by the model.

In this project, we create a more sophisticated attack based around the work by Papernot et al.[13] Named Bloodhound, this attack attempts to train a "bloodhound" model that emulates the decision borders of the target model. White-box attacks are performed on the bloodhound model, and because of the transferability in attacks, these attacks are also effective against the target model. Transferability is the concept that attacks generated on a neural network will cause misclassification in a separate neural network, even when the models' architectures are different.[9] To train this model, we used a set of 54,503 unlabeled sound files of the same format and similar classification distribution as the original labeled dataset. This large dataset was originally provided as a way to test model accuracy for a competition held by the dataset's creators.

To emulate the target model, the attacker passes the unlabeled data, soft-labeling it with the target model's output (Fig. 7). The image and label are then passed to the new model to train. We confirmed transferability between architectures by testing FGSM images trained with AlexNet on ResNet18,[5] a newer CNN with significantly more layers than AlexNet, and the attacks performed nearly identically. Therefore, the assumption is made that Bloodhound would work against a model of unknown architecture. This attack is still not completely realistic, for creating such a large dataset would be expensive, and the thousands of queries on the target model would be easily detectable. Future work will adjust Bloodhound to generate images to isolate the target model's decision boundaries in fewer queries, following the original example.[13]

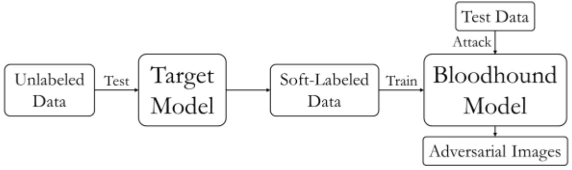


Figure 7: Bloodhound Visualized

When testing attacks generated on this new model, we found that high-epsilon FGSM attacks successfully disrupted the model, albeit at a lower strength than direct white-box attacks. However, we found that CW and DeepFool, attacks that attempt to barely cross decision borders for minimal perturbations, had no effect on the target model. This suggests that the two models do not have the same decision borders. We theorize that the first model is not trained on a large or diverse enough dataset to define intelligent decision borders, and the existing hyperplanes are somewhat arbitrary, for a different model (the bloodhound) can be trained to have similar accuracy yet different decision borders.

C. Attack Results

Figure 8 shows the performance of each attack compared to the original model accuracy. The same set of data was used for all tests, perturbed in real-time. DeepFool was the most effective attack studied, for it reduced accuracy by 40% more than the next best attack while remaining unnoticeable.

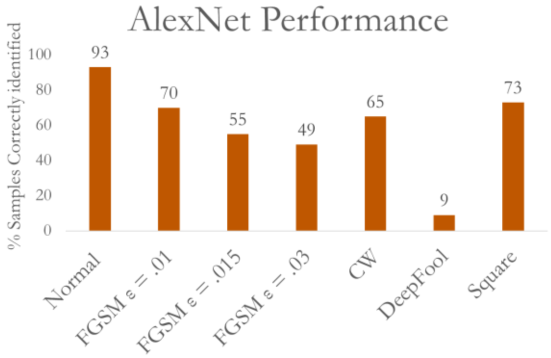


Figure 8: Attack performance on AlexNet Architecture

As seen in Fig. 9, the Bloodhound attacks are very effective on the bloodhound model, but only high-epsilon FGSM attacks transfer to the target model. CW was excluded from bloodhound tests because of its high computation time and near-identical results to DeepFool in small-scale testing.

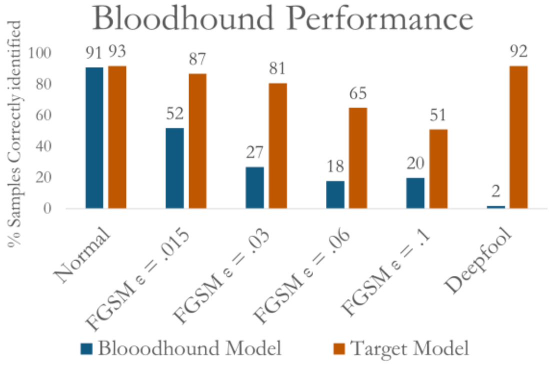


Figure 9: Bloodhound Attack Statistics

IV. Defenses

Even at low epsilon values, these attacks prove highly effective, degrading the AlexNet model's performance to near-random chance while remaining very similar to the original sample, even visually identical for smaller ε values. Therefore, different proposals of adversarial defenses were investigated and experimented relating to data augmentation and pre-processing prior to network input.

A. Adversarial Training

Adversarial training augments the model training pipeline with perturbed data to encourage robustness against similar attacks.[9] A baseline model is trained with standard data, then adversarial perturbations are developed to attack this model. These adversarial images are then used to train a robust model.

A problem with this methodology was quickly discovered: the model is only robust to perturbations included in the training. To solve this, the AlexNet model was trained on adversarial data generated on the fly, varying the ε values in a uniform distribution between 0 and .03. By re-generating adversarial images between epochs of the training sequence, we create an arms race between the model and generator, each attempting to outdo each other every epoch. This process is computationally intensive and time consuming but greatly increases robustness.

B. Low-Pass/Data Smoothing/Gaussian Blur

Most white-box attack perturbations were observed to have large fluctuations between pixels, especially FGSM. This allows for a low pass filter defense, the Gaussian blur (Fig. 10), on the spectrogram data in order to smooth out the small pixel changes from the attack. The Gaussian blur therefore hinders white-box attacks utilizing the gradient, as highly variable gradient-based perturbations are filtered out.

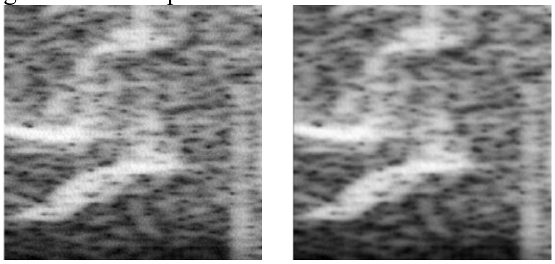


Figure 10: Perturbed image transformed with Gaussian blur

Inspired by the SHIELD adversarial defense,[14] which utilizes lossy JPEG compression to have a similar effect, this defense adds a gaussian blur to all data fed into the model, creating a "vaccinated" model that is unaffected by subtle changes to inputs. The blur operation is processed through the data loader, allowing for combination with adversarial training defense. We found that minor blurring drastically reduced the attacks' efficacy while causing negligible accuracy loss from the baseline performance.

Gaussian blur is implemented with a strong and weak variation. The weak variation operates on a 3x3 image kernel convolution with $\sigma = 1.5$ while the strong variation operates on a 5x5 kernel, $\sigma = 3$ in order to evaluate the tradeoff between gaussian blur strength's effectiveness against adversarial attacks and collateral accuracy loss against baseline performance.

C. Defenses Results

Both Gaussian blur and adversarial training defenses are shown to be highly effective at reducing attack severity against the model while maintaining normal accuracy within 6% on adversarial training and 1% with Gaussian blur.

The FGSM 0.3 attack remains most damaging against all defenses, including combinations of adversarial training and blurring. Both Gaussian blur and FGSM training defenses individually had parts of the receiver operating characteristics where the attack still causes the classifier to have cases fall below random chance (black line in Fig. 11).

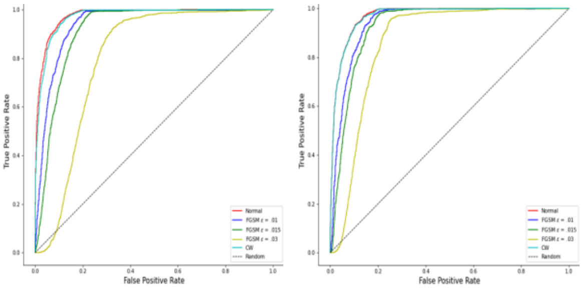


Figure 11: ROC curves for weak Gaussian Blur (left) and adversarial training (right). Adversarial training outperforms Gaussian blur.

A stronger Gaussian blur proved to be more effective than the weak blur, remaining comparable at the baseline and achieving higher accuracies across all attacks. Additionally, Gaussian blur was shown to have negligible effect on computational cost and training time while adversarial training increased training time by a noticeable amount as shown in figure 12.

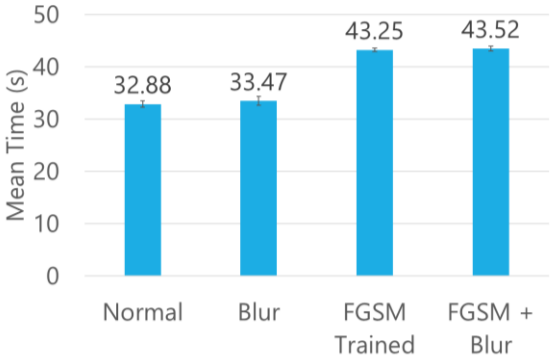


Figure 12: Training time comparison of defenses.

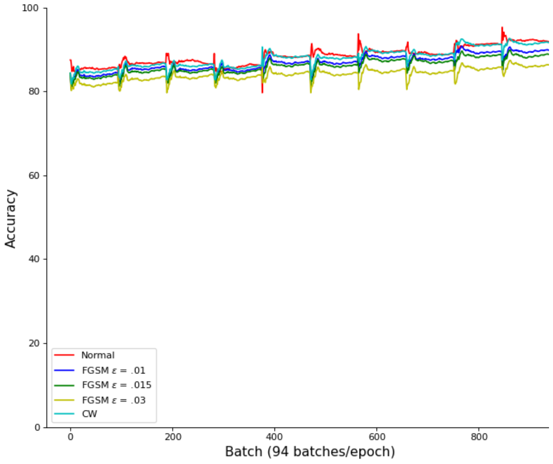


Figure 13: Combined FGSM + 2X (strong) Gaussian Blur.

Overall, combining adversarial training and strong blur yields the best results, with consistently high accuracies between the baseline (92%) and the

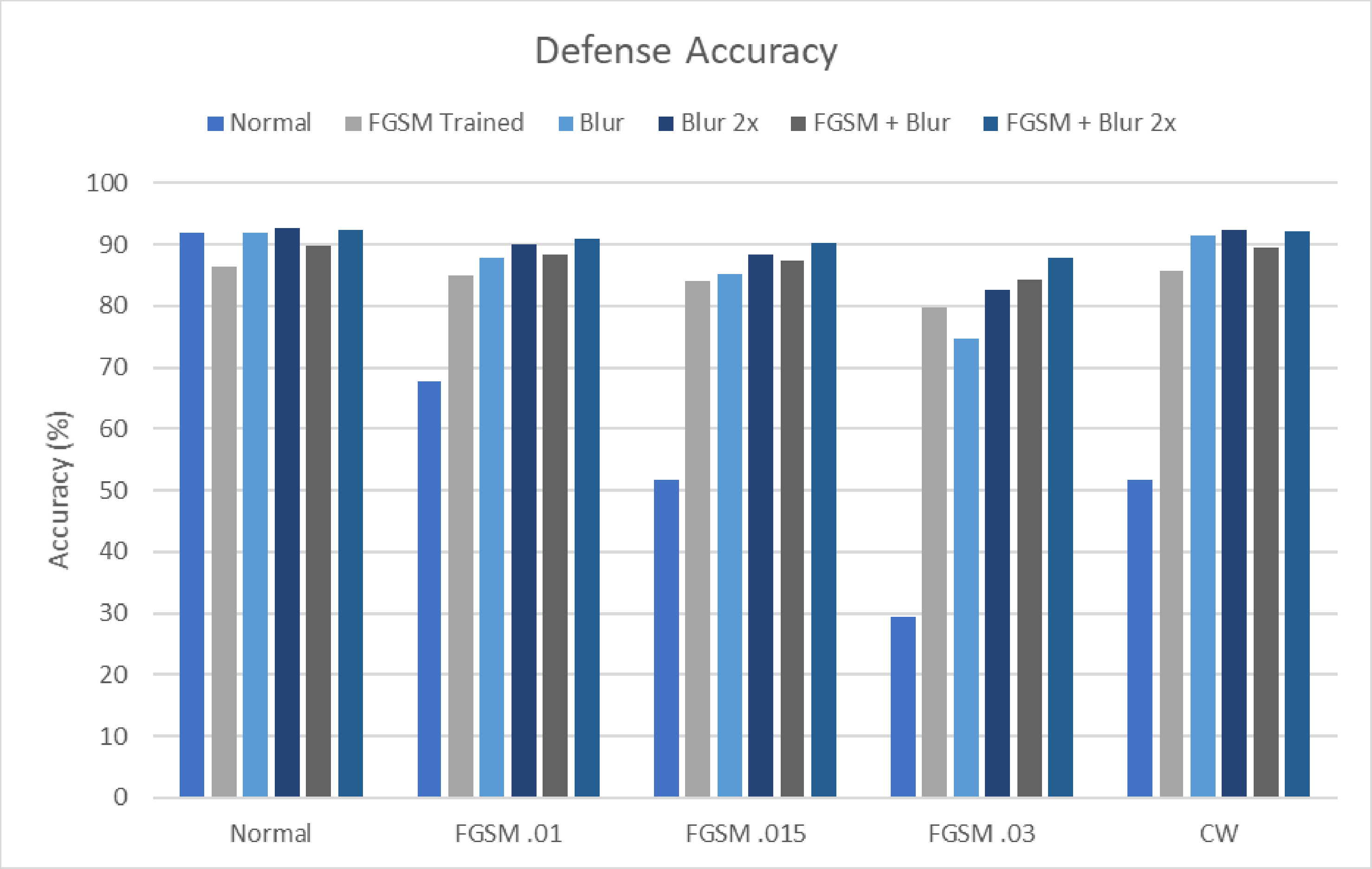


Figure 14: Overall defenses accuracy comparison against multiple attacks

largest attacks (86%) (Fig. 13). Figure 14 shows the accuracy of each defense on FGSM and CW attacks.

V. Conclusions

A. Results

Likely because of the small dataset provided, small neural networks like AlexNet perform best at this binary classification problem. Deeper networks like ResNet and VGG tend to overfit and are brittle to attacks.

Most attacks tested in this project caused the model to predict negative for the majority of samples, not rising to the level of full targeted disruption, but rendering the model useless. However, by using high-epsilon FGSM attacks and DeepFool, we were able to create targeted misclassifications. This means that we exhibited full control over the model's outputs, a much more powerful position than with the other attacks. Square and Bloodhound, the black-box attacks tested, successfully disrupted the model without internal knowledge when high levels of perturbation were permitted, and with a more sophisticated model trained on a larger dataset, the Bloodhound attack would likely successfully attack with delicate perturbations.

Defenses proved capable of mitigating white-box attacks without sacrificing considerable accuracy, precision, or recall against unmodified data. Combining multiple defenses also enhanced effectiveness without causing conflicts.

B. Limitations

Through analysis of the dataset, three problems emerge: The dataset is too small to create significant decision borders inside the model's parameters, there are many more negative samples than positive, and the dataset contains mislabeled samples.

The labeled dataset contains 30,000 samples. While this amount is enough to train a reasonably accurate model, it is not enough to fully teach the subtle differences between whale calls. When testing the Bloodhound attack strategy (see 3.B), we found that DeepFool, an attack designed to barely cross the nearest decision boundary, performed strongly on the bloodhound model but had no effect on the target model. This suggests that the boundaries between models are arbitrary and need more data to be accurately fleshed out.

To address this problem moving forward, research will shift towards larger dataset containing real-world interference such as passing boats and other mammal sounds. Additionally, the dataset will be bootstrapped, or artificially expanded, by duplicating training examples with added noise and truncating samples in both the frequency and time domain. By adding adversarial defense training images as described in 4.A and 4.B, this project takes a first step towards expanding the dataset. Testing showed that these techniques increased accuracy and helped the model to generalize.

The dataset contains an imbalance in the number of named samples, containing only ~7000 positives and ~23,000 negatives. As detailed at the end of 1.A, this disparity was successfully addressed by weighting samples differently in training depending on their label.

The final problem with the data is systemic and out of the scope of this project. The dataset was designed to be used to distinguish North Atlantic right whale up-calls from noise and other similar-sounding mammal calls, such as humpback whale calls. Oftentimes, the 2-second samples are too short to correctly distinguish these calls (Fig. 15), thus confusing our model with seemingly missed calls. Additionally, the creators of the dataset have announced that some samples are overtly mislabeled.

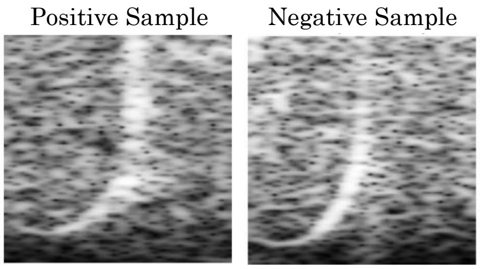


Figure 15: While these spectrograms are labeled differently, they look similar enough to confuse the model

C. Future Work

We are currently expanding the project to create a generalized sound classifier that is robust against adversarial attacks. The long-term goal is a real-time classifier that passively monitors a section of the ocean, grabbing and classifying sound samples of underwater mammals and ships.

Future research will directly alter sound samples to disrupt the model rather than arbitrary changes to the generated spectrograms.

Acknowledgements

We would like to thank Reid Wyde and Scott Johnston, our supervisors for this project, for their support and guidance. We would also like to thank Anna Chaney, our unofficial supervisor, for answering our questions and pointing us to helpful resources. Finally, we would like to thank Hector Gonzalez, our Honors Scholar, for providing direction in our project and helping to debug our code.

References

[1] C. Szegedy et al, “Intriguing properties of neural networks,” *arXiv* **arXiv:1312.6199** (2013).

[2] Team Panda, “Class 1: Intro to Adversarial Machine Learning,” https://secml.github.io/class1/ (Accessed 28 Jul. 2021)

[3] Kaggle, “The Marinexplore and Cornell University Whale Detection Challenge,” https://www.kaggle.com/c/whale-detection-challenge (Accessed August 2, 2021)

[4] A. Krizhevsky, I. Sutskever and G. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Association for Computing Machinery* **volume 60** (2017)

[5] He et al, “Deep Residual Learning for Image Recognition,” *arXiv* **arXiv:1512.03385** (2015).

[6] Howard et al, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications” *arXiv* **arXiv:1704.04861** (2017).

[7] K Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv* **arXiv:1409.1556** (2014).

[8] A. Anwar, “Difference between AlexNet, VGGNet, ResNet, and Inception, https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96 (Accessed 9 Aug 2021).

[9] I. J. Goodfellow, J. Shlens and C. Szegedy, “Explaining and Harnessing Adversarial Examples,” *arXiv* **arXiv:1412.6572** (2014).

[10] N. Carlini and D. Wagner, “Towards Evaluating the Robustness of Neural Networks,” *arXiv* **arXiv:1608.04644** (2016).

[11] S. Moosavi-Dezfooli, A. Fawzi and P. Frossard, "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks," *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2574-2582 (2016), doi: 10.1109/CVPR.2016.282.

[12] M. Andriushchenko et al, “Square Attack: a query-efficient black-box adversarial attack via random search,” *arXiv* **arXiv:1912.00049** (2020).

[13] N. Papernot et al, "Practical Black-Box Attacks against Machine Learning," *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 506-519 (2017), doi: 10.1145/3052973.3053009.

[14] N. Das et al, "SHIELD: Fast, Practical Defense and Vaccination for Deep Learning using JPEG Compression," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 196-204 (2018), doi: 10.1145/3219819.3219910