Robust Adversarial Image Detection in Classification

Arti Jain, Grant Ovsepyan, Nithanth Ram

Problem Statement

- Adversarial attacks on machine learning models can have catastrophic effects
  - Ex. Autonomous vehicle system trying to perform traffic sign classification
  - White-box attacks (access to model's gradients) and black-box attacks (no access to model)
- Investigate various conditioning techniques to explore the possibility of strengthening classification at inference time against white-box and black-box image perturbations

Previous Work

Color and Edge-Aware Adversarial Image Perturbations [1]

- New easy-to-compute subtle perturbations involving regions of smoothness and color contrast

Adversarial Training for Free! [3]

- Add semi-supervised virtual adversarial noise to the input during training, acting as a regularizer

n-ML: Mitigating Adversarial Examples via Ensembles of Topologically Manipulated Classifiers [2]

- Train multiple topologies so that ensembling these networks together provides more defense against adversarial examples

Dataset

Dataset: GTSRB (German Traffic Sign Recognition Benchmark)
- 43 classes (speed limit, stop sign, pedestrian crossing, etc.)
- 39,210 training images

White Box Attacks
- FGSM (Fast Gradient Sign Method)
  - Fast and efficient method to generate attack
- PGD (Projected Gradient Descent)
  - Worst case boundaries of the model performance

Black Box Attack:
- Gaussian Blur
  - mimics real world scenario -> out of focus images

Original Image

FGSM

PGD

Gaussian Blur

Our Approach

- Our approach was to experiment with image manipulation techniques to observe the change in misclassification from a perturbed dataset
  - Some of these processing techniques included utilizing canny edge maps, depth maps, contrast/light enhancement, de-blurring, de-noising

- We tried 3 methods:
  1. Concatenating canny edge maps to the input data and then passing it through our trained model
  2. Using different image processing techniques on the perturbed data and then running inference
  3. Ensembling separate classifiers/sub-neural networks

Experiment Framework

- Fine-tune existing pre-trained image classifier models on traffic sign dataset
- Perform adversarial perturbations to dataset and measure degradation of model efficacy
- Tried mitigation technique
  - Concatenating Canny Edge image to original image
  - Unblurring/denoising image (MPRNet), contrast enhancement, K-means clustering, based on the color similarity
  - Ensembling and stacking models to get more robust classification
- Use F1-score and accuracy to track the performance across experiments

Training

- Base Models: ResNet50, InceptionNetV3, MobileNetV3
  - ResNet50 fine-tuned ~ 76% test accuracy, 0.73 test F1-score
  - InceptionV3 fine-tuned ~ 68% test accuracy, 0.70 test F1-score
  - MobileNetV3 fine-tuned ~ 99.3% test accuracy, 0.99 F1-score

- Proceeding with MobileNetV3 as baseline fine-tuned model
- Apply adversarial perturbations with fixed parameters for control
  - Gaussian Blur: kernel_size=5, sigma=2 (severe blur from visual inspection)
  - FGSM attack epsilon = 0.1, (1 iteration) (significant noise perturbation, additional noise would perturn image too much)
  - PGD attack epsilon = 0.1, alpha= 0.04, num_iteration = 10 (less noise perturbation than FGSM, noise density between original unperturbed image and FGSM image)

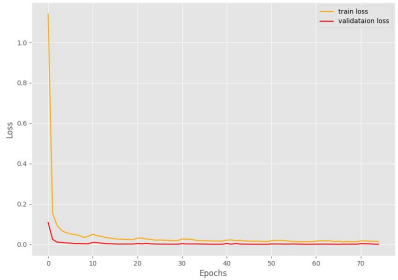
.

- Decided to manually sort the classes for further inference using most readable images taken under reasonable light conditions
- Selected classes are: **0,1,2,11,12,14,16,28,34,35,39 (3810 images)**
- Baseline test accuracies on the adversarial images using MobileNetV3 on the selected classes are
  - Gaussian Blur: 89.61%
  - FGSM attack: 67.14%
  - PGD attack: 77.27%

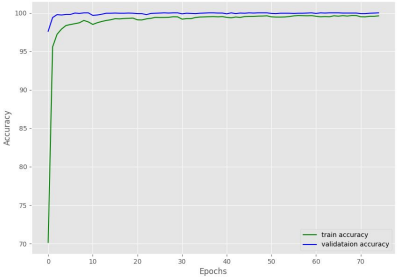
Training and Validation Curves

MobileNetV3:

training and validation loss

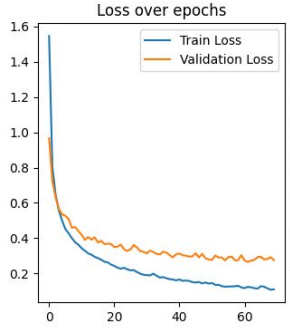


training and validation accuracies

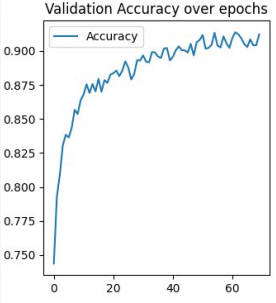


ResNet50:

training and validation loss

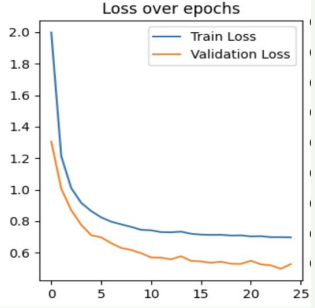


validation accuracy

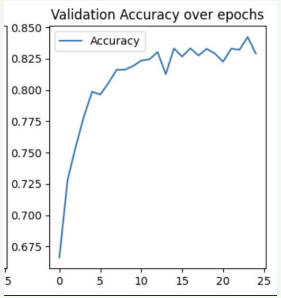


InceptionNetV3:

training and validation loss



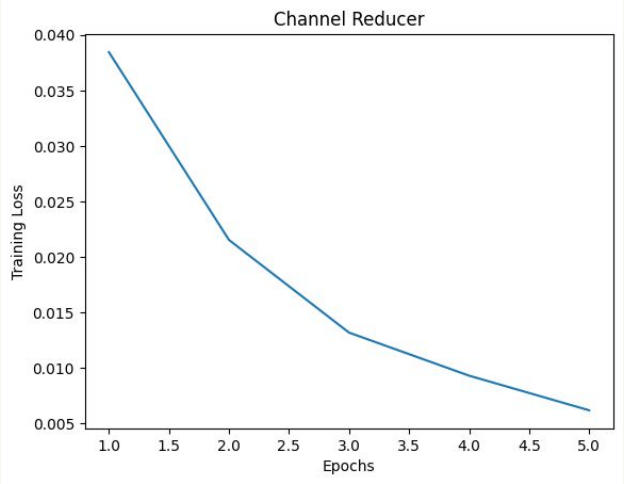
validation accuracy



Solution Space (MobileNetV3)

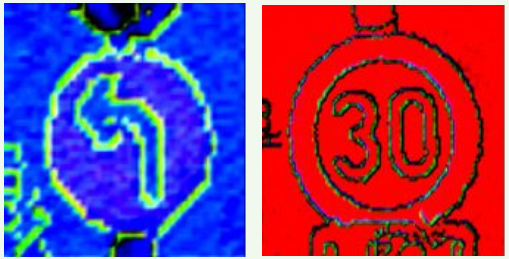
- Concatenate edge image and normal image before performing inference
  a. Expand 3 channel image to 4 channel and pass through fine tuned model
  b. See if model can't already judge augmented inputs with its current weights

- Trained a simple CNN (one layer) to convert 4 channels into 3
  - Used training data to create pairs of canny edge maps as the input, the original image as the label
  - 5 epochs, MSE Loss, LR = 0.001

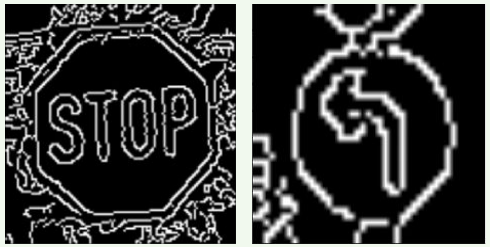


Solution Space (MobileNetV3)

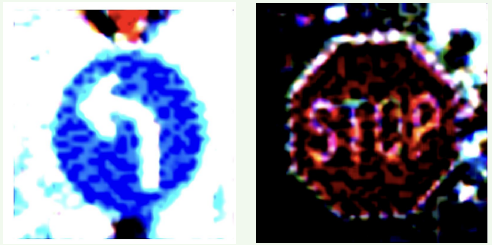
Perturbed Data with FGSM attack



Reducing from 4 channels to 3 channels with random CNN weights



Canny images on perturbed data



Reducing from 4 channels to 3 channels with trained CNN

Solution Space (MobileNetV3)

	Test Accuracy	F1 Score
Baseline with FGSM Attack	97.35%	0.986
With Canny Edge and trained CNN	75.76%	0.82
With Canny Edge and random CNN weights	30.48%	0.29

	Test Accuracy	F1 Score
Baseline with Gaussian Blur Attack	89.61%	0.92
With Canny Edge and trained CNN	86.726%	0.897

	Test Accuracy	F1 Score
Baseline with PGD Attack	77.27%	0.84
With Canny Edge and trained CNN	81.42%	0.86

Solution Space (MobileNetV3)

1) Denoising/Deblurring using MPRNet (off-shelf)
   a) Intuitively, if FGSM and PGD add the noise to the image, then denoising those images should reverse noise addition. Similarly, deblurring should reverse the effects of Gaussian Blur

   MPRNet was chosen because it utilizes multiple stages to progressively restore the image focusing on different aspects of restoration:

   1) Encoder – Decoder Network: large receptive fields essential for understanding the overall image context.
   2) Original Resolution Network: preserve the fine details necessary for high-quality restoration.
   3) SAM: Supervised Attention Module (SAM) is used between each stage to refine the processed features is used to guide the attention mechanism
   4) CSFF: Cross Stage Feature Fusion Technique, that ensures that integrates the features from different stages preventing the loss in details

2) Denoising/Deblurring using MPRNet + contrast enhancement
   a) After inspecting some images, it was noted that some images might not have defined sign boundaries that blend in with environment, so contrast enhancement was introduced.


baseline_original_images
baseline_gaussian_blur
de-gaussian_blur
de-gaussian_blur_light
50
20
STOP


baseline_original_images
baseline_fgsm_attack
de-fgsm_attack
de-fgsm_attack_light
50
50
50
20
20
20
STOP
STOP
STOP
STOP


baseline_original_images
baseline_pgd_attack
de-pgd_attack
de-pgd_attack_light
50
20
STOP

Test accuracies on 3810 images on the selected classes (0,1,2,11,12,14,16,28,34,35,39)

Image Set	Test Accuracy	F1 Score
Original Images	99.30%	0.9953
Gaussian Blur Baseline	89.61%	0.9196
Deblurred (MPRNet)	87.74%	0.8998
Deblurred (MPRNet) + Contrast enhancement	88.35%	0.8836
FGSM Baseline	67.14%	0.7540
Denoised FGSM (MPRNet)	78.11%	0.8566
Denoised FGSM (MPRNet) + Contrast enhancement	82.23%	0.8737
PGD Baseline	77.27%	0.8431
Denoised PGD (MPRNet)	83.76%	0.8953
Denoised PGD (MPRNet) + Contrast enhancement	87.16%	0.9115

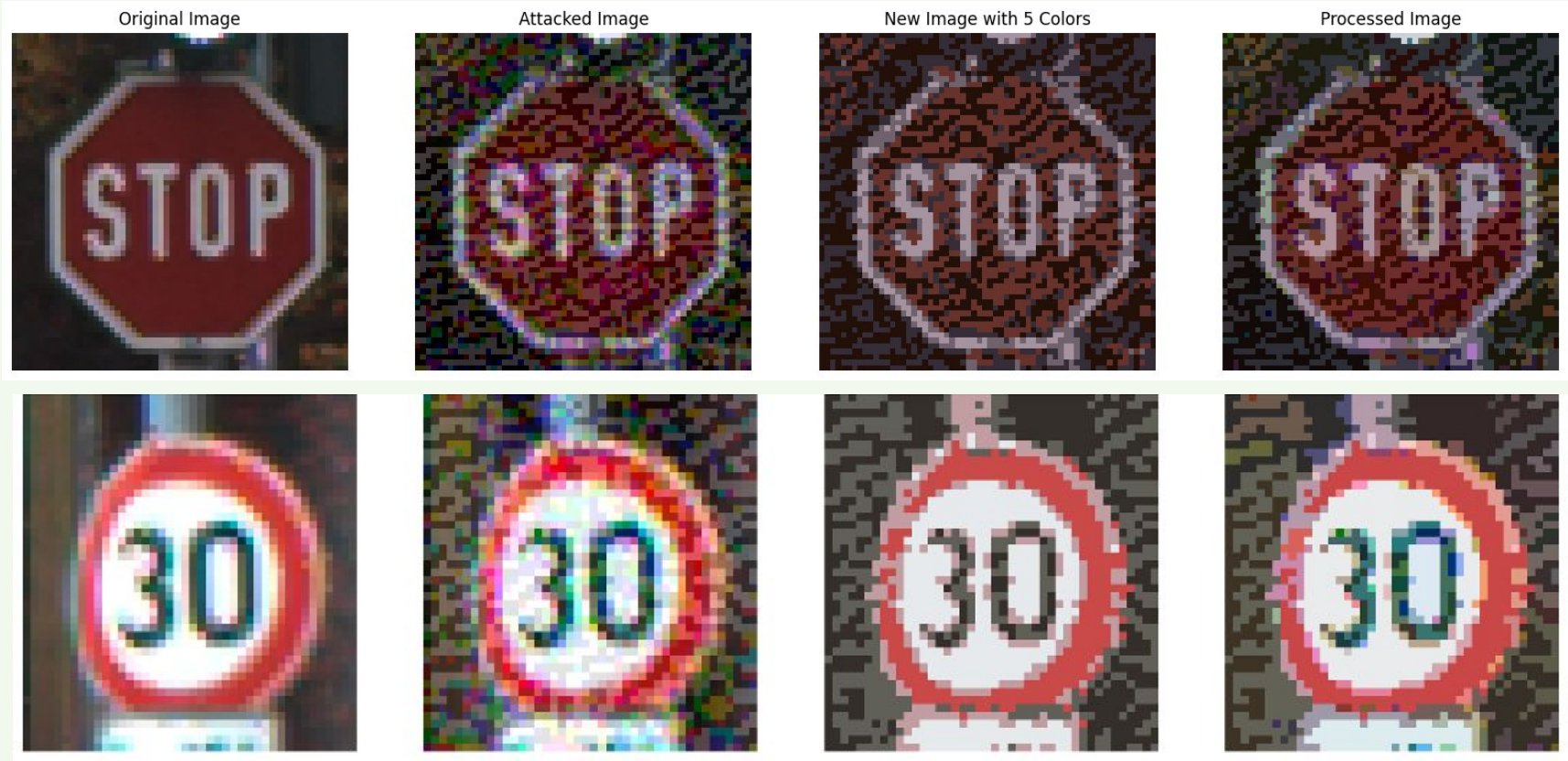
Solution Space (MobileNetV3)

1) Manual clustering and pseudo-denoising based on k-Means clustering
    a) Why?
        i) From previous experiments the denoising approach improved the test accuracy on the PGD and FGSM images.

    b) How?
        i) Empirically determined most optimal k value = 5 that corresponds to number of most dominant colors in the image.
        ii) Segment the images based on 5 dominant colors
        iii) For each segmented group (not color, since there will be multiple groups for the same color):
            1) Average the pixel values on the adversarial image for each group
            2) Replace the values of that group with an averaged value

Solution Space (MobileNetV3)

1) Color Averaging: pseudo-denoising based on k-Means clustering

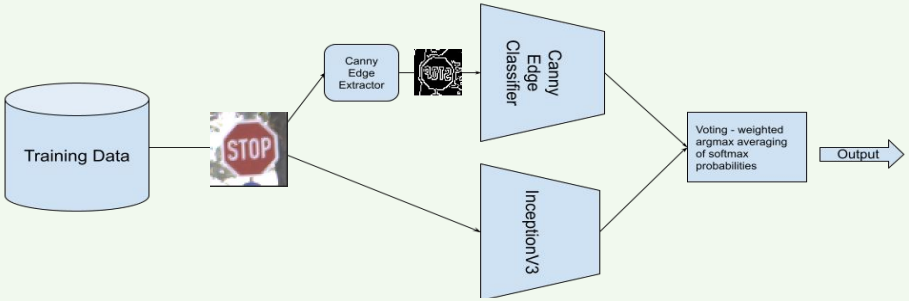


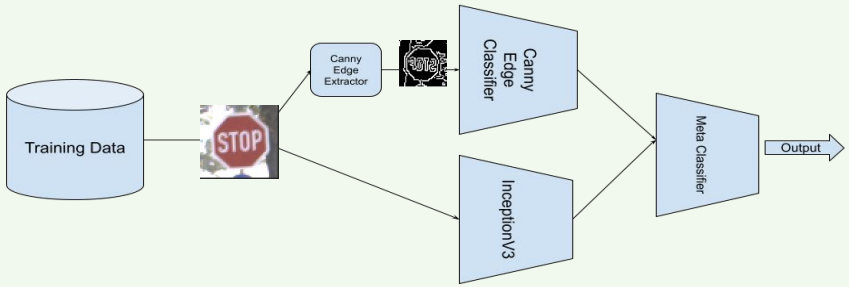
Test accuracies on 3810 images on the selected classes (0,1,2,11,12,14,16,28,34,35,39)

Image Set	Test Accuracy	F1 Score
Original Images	99.30%	0.9953
Original Images + Color Averaging	95.88%	0.9708
Gaussian Blur Baseline + Color Averaging	72.89%	0.7766
Deblurred (MPRNet) + Color Averaging	79.11%	0.8344
Deblurred (MPRNet) + Contrast enhancement + Color Averaging	80.34%	0.8412
FGSM Baseline + Color Averaging	69.19%	0.7677
Denoised FGSM (MPRNet) + Color Averaging	70.76%	0.7875
Denoised FGSM (MPRNet) + Contrast enhancement + Color Averaging	70.84%	0.7719
PGD Baseline + Color Averaging	78.85%	0.8498
Denoised PGD (MPRNet) + Color Averaging	77.09%	0.8384
Denoised PGD (MPRNet) + Contrast enhancement + Color Averaging	76.38%	0.8178

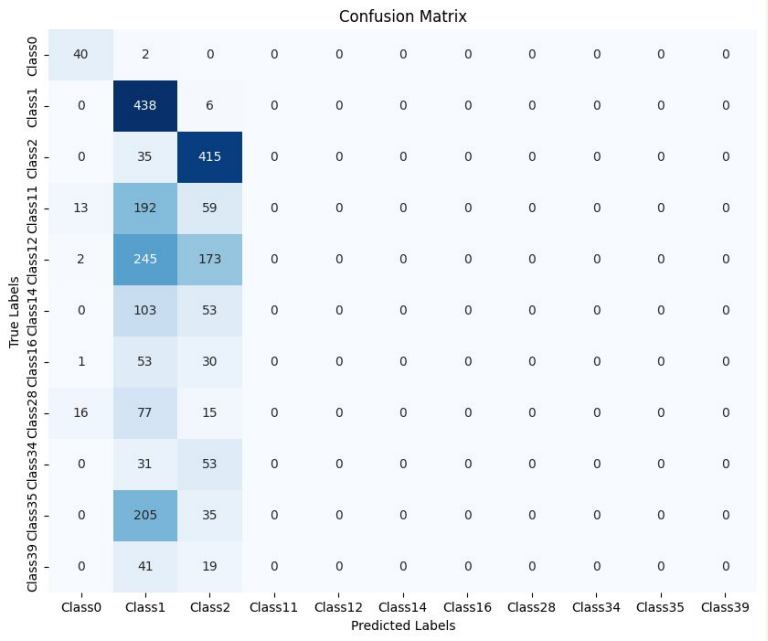
Solution Space (InceptionV3)

- Stacking/Ensembling classifiers for edge image and normal image
  - Weighted voting - averaging the softmax probabilities from both and applying a weighting based on which classifier is better performant
  - Training a meta-classifier - overall neural network that has the independent image networks as sub networks





- Stacking/Ensembling failed - worse than random guessing ~ 45% accuracy
  - Canny Edge classifier had poor performance
  - Quality of edge extracted images for a lot of classes was poor - no edges detected



Results & Conclusion

Given the results, it is clear that FGSM had the most impact (32.16% drop) on the baseline test accuracy, then PGD (22.03% drop) and then Gaussian Blur (9.69% drop).

Gaussian Blur:

- Deblurring and Contrast Enhancement did not significantly changed the test accuracy (fluctuates within ~1%).
- Canny Edge Concatenation did not significantly change the test accuracy (fluctuates within 3.22%)

FGSM:

- Denoising improved test accuracy by 10.97% & F1 score improved by around 0.1
- Contrast enhancement improved accuracy by 4.12% and F1 score improved by 0.12
- Canny Edge Concatenation decreased test accuracy by 22.17% & F1 score decreased by 0.16

PGD:

- Denoising improved test accuracy by 6.49% and F1 score improved by 0.05
- Contrast enhancement improved accuracy by 3.4% and F1 score improved by 0.07
- Canny Edge Concatenation improved accuracy by 5.37% and F1 score improved by 0.02

- Techniques such as color contrast, ensembling were not successful mitigation techniques

Results & Conclusion

- Gaussian blur attacks are the most difficult to protect against at inference time
  - Black box attacks are more difficult to protect against
- Multiple techniques were able to improve accuracy on a PGD attack
  - Maybe it was a milder attack than FGSM (not entirely sure)
- The de-noising and contrast enhancement were effective against FGSM but canny edge concatenation was not
  - De-noising is directly combatting FGSM while canny edge is not
- Future works include:
  - Combining the successful mitigation techniques
  - Trying ensemble methods with features other than canny edge
  - Trying with a different dataset to ensure that results are consistent
  - Try mitigation techniques on various strengths of attacks
- It’s difficult to improve performance on inference time alone, but not eager to rule it out

Thank you!

Thank you to our Professors and TA mentor, Weiran, for all your help and guidance!

Resources

[1] Bassett, R. (2021). Color and Edge-Aware Adversarial Image Perturbations.

[2] Hosseini, H., Xiao, B., & Poovendran, R. (2019). n-ML: Mitigating Adversarial Examples via Ensembles of Topologically Manipulated Classifiers.

[3] Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., ... & Goldstein, T. (2019). Adversarial training for free!

[4] Zamir S., Arora A., et al., (2021). CVPR. Multi-Stage Progressive Image Restoration.

[5] Rath S. (2022), Traffic Sign Recognition using PyTorch and Deep Learning. Debugger Cafe. https://debuggercafe.com/traffic-sign-recognition-using-pytorch-and-deep-learning/

[6] Color-Extraction-and-Image-Segmentation.(2020). https://github.com/Ronny-22-Code/Color-Extraction-and-Image-Segmentation.git

Extra Slides

Supporting materials (MobileNetV3)

```
baseline_original_images:Total number of test images: 3810
baseline_original_images:Total correct predictions: 3783
baseline_original_images:Accuracy: 99.291
Average FPS: 7.297
```

```
baseline_gaussian_blur:Total number of test images: 3810
baseline_gaussian_blur:Total correct predictions: 3414
baseline_gaussian_blur:Accuracy: 89.606
Average FPS: 8.765
baseline_pgd_attack:Total number of test images: 3810
baseline_pgd_attack:Total correct predictions: 2944
baseline_pgd_attack:Accuracy: 77.270
Average FPS: 8.290
```

```
baseline_fgsm_attack:Total number of test images: 3810
baseline_fgsm_attack:Total correct predictions: 2558
baseline_fgsm_attack:Accuracy: 67.139
Average FPS: 8.629
```

```
de-gaussian_blur:Total number of test images: 3810
de-gaussian_blur:Total correct predictions: 3343
de-gaussian_blur:Accuracy: 87.743
Average FPS: 7.327
de-pgd_attack:Total number of test images: 3810
de-pgd_attack:Total correct predictions: 3191
de-pgd_attack:Accuracy: 83.753
Average FPS: 7.455
de-fgsm_attack:Total number of test images: 3810
de-fgsm_attack:Total correct predictions: 2976
de-fgsm_attack:Accuracy: 78.110
Average FPS: 7.298
```

```
de-gaussian_blur_light:Total number of test images: 3810
de-gaussian_blur_light:Total correct predictions: 3366
de-gaussian_blur_light:Accuracy: 88.346
Average FPS: 10.029
```

```
de-pgd_attack_light:Total number of test images: 3810
de-pgd_attack_light:Total correct predictions: 3321
de-pgd_attack_light:Accuracy: 87.165
Average FPS: 9.774
de-fgsm_attack_light:Total number of test images: 3810
de-fgsm_attack_light:Total correct predictions: 3133
de-fgsm_attack_light:Accuracy: 82.231
Average FPS: 9.641
```

Supporting materials (MobileNetV3)

When color averaging is applied

```
baseline_original_images_color_avg:Total number of test images: 3810
baseline_original_images_color_avg:Total correct predictions: 3653
baseline_original_images_color_avg:Accuracy: 95.879
Average FPS: 10.068
```

```
de-gaussian_blur_color_avg:Total number of test images: 3810
de-gaussian_blur_color_avg:Total correct predictions: 3014
de-gaussian_blur_color_avg:Accuracy: 79.108
Average FPS: 7.603
```

```
de-pgd_attack_color_avg:Total number of test images: 3810
de-pgd_attack_color_avg:Total correct predictions: 2937
de-pgd_attack_color_avg:Accuracy: 77.087
Average FPS: 8.081
```

```
de-fgsm_attack_color_avg:Total number of test images: 3810
de-fgsm_attack_color_avg:Total correct predictions: 2696
de-fgsm_attack_color_avg:Accuracy: 70.761
Average FPS: 8.101
```

```
baseline_gaussian_blur_color_avg:Total number of test images: 3810
baseline_gaussian_blur_color_avg:Total correct predictions: 2777
baseline_gaussian_blur_color_avg:Accuracy: 72.887
Average FPS: 8.154
```

```
baseline_pgd_attack_color_avg:Total number of test images: 3810
baseline_pgd_attack_color_avg:Total correct predictions: 3004
baseline_pgd_attack_color_avg:Accuracy: 78.845
Average FPS: 7.634
```

```
baseline_fgsm_attack_color_avg:Total number of test images: 3810
baseline_fgsm_attack_color_avg:Total correct predictions: 2636
baseline_fgsm_attack_color_avg:Accuracy: 69.186
Average FPS: 8.139
```

```
de-gaussian_blur_light_color_avg:Total number of test images: 3810
de-gaussian_blur_light_color_avg:Total correct predictions: 3061
de-gaussian_blur_light_color_avg:Accuracy: 80.341
Average FPS: 7.745
```

```
de-pgd_attack_light_color_avg:Total number of test images: 3810
de-pgd_attack_light_color_avg:Total correct predictions: 2910
de-pgd_attack_light_color_avg:Accuracy: 76.378
Average FPS: 9.964
```

```
de-fgsm_attack_light_color_avg:Total number of test images: 3810
de-fgsm_attack_light_color_avg:Total correct predictions: 2699
de-fgsm_attack_light_color_avg:Accuracy: 70.840
Average FPS: 10.125
```

Supporting materials (MobileNetV3)

F-1 scores

```
(base) gman@Grants-MacBook-Air src % python f_1.py baseline_fgsm_attack

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_ut
s deprecated since 0.13 and may be removed in the future, please use 'wei
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_ut
t enum or `None` for 'weights' are deprecated since 0.13 and may be remov
to passing `weights=None`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.7540
```

```
(base) gman@Grants-MacBook-Air src % python f_1.py baseline_fgsm_attack_color_avg

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:20
etrained' is deprecated since 0.13 and may be removed in the future, please use 'w
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:22
han a weight enum or `None` for 'weights' are deprecated since 0.13 and may be rem
ehavior is equivalent to passing `weights=None`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.7677
```

```
(base) gman@Grants-Air src % python f_1.py baseline_pgd_attack_color_avg

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.
s deprecated since 0.13 and may be removed in the future, please use 'weights
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.
t enum or `None` for 'weights' are deprecated since 0.13 and may be removed i
to passing `weights=None`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.8498
```

```
(base) gman@Grants-Air src % python f_1.py de-fgsm_attack

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:20
etrained' is deprecated since 0.13 and may be removed in the future, please use 'w
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:22
han a weight enum or `None` for 'weights' are deprecated since 0.13 and may be rem
ehavior is equivalent to passing `weights=None`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.8566
```

```
(base) gman@Grants-Air src % python f_1.py de-gaussian_blur_light

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.
s deprecated since 0.13 and may be removed in the future, please use 'weights
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.
t enum or `None` for 'weights' are deprecated since 0.13 and may be removed i
to passing `weights=None`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.8836
```

```
(base) gman@Grants-Air src % python f_1.py de-gaussian_blur_light_color_avg

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:20
etrained' is deprecated since 0.13 and may be removed in the future, please use 'w
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:22
han a weight enum or `None` for 'weights' are deprecated since 0.13 and may be rem
ehavior is equivalent to passing `weights=None`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.8412
```

Supporting materials (MobileNetV3)

F-1 scores

```
(base) gman@Grants-MacBook-Air src % python f_1.py baseline_gaussian_blur
[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:208:
ning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
please use 'weights' instead.
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:223:
ning: Arguments other than a weight enum or `None` for 'weights' are deprecated sinc
nd may be removed in the future. The current behavior is equivalent to passing `weig
`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.9196
```

```
(base) gman@Grants-MacBook-Air src % python f_1.py baseline_gaussian_blur_color_avg

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:208:
ning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
please use 'weights' instead.
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:223:
ning: Arguments other than a weight enum or `None` for 'weights' are deprecated sinc
nd may be removed in the future. The current behavior is equivalent to passing `weig
`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.7766
```

```
(base) gman@Grants-Air src % python f_1.py de-fgsm_attack_color_avg

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:208:
ning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
please use 'weights' instead.
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:223:
ning: Arguments other than a weight enum or `None` for 'weights' are deprecated sinc
nd may be removed in the future. The current behavior is equivalent to passing `weig
`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.7875
```

```
(base) gman@Grants-Air src % python f_1.py de-fgsm_attack_light

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:208:
ning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
please use 'weights' instead.
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:223:
ning: Arguments other than a weight enum or `None` for 'weights' are deprecated sinc
nd may be removed in the future. The current behavior is equivalent to passing `weig
`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.8737
```

```
(base) gman@Grants-Air src % python f_1.py de-pgd_attack

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:208:
ning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
please use 'weights' instead.
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:223:
ning: Arguments other than a weight enum or `None` for 'weights' are deprecated sinc
nd may be removed in the future. The current behavior is equivalent to passing `weig
`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.8953
```

```
(base) gman@Grants-Air src % python f_1.py de-pgd_attack_color_avg

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:208:
ning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
please use 'weights' instead.
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:223:
ning: Arguments other than a weight enum or `None` for 'weights' are deprecated sinc
nd may be removed in the future. The current behavior is equivalent to passing `weig
`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.8384
```

Supporting materials (MobileNetV3)

F-1 scores

```
(base) gman@Grants-MacBook-Air src % python f_1.py baseline_original_images

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:208:
ning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
please use 'weights' instead.
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:223:
ning: Arguments other than a weight enum or `None` for 'weights' are deprecated sinc
nd may be removed in the future. The current behavior is equivalent to passing `weig
`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.9953
```

```
(base) gman@Grants-Air src % python f_1.py de-fgsm_attack_light_color_avg

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:208:
ning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
please use 'weights' instead.
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:223:
ning: Arguments other than a weight enum or `None` for 'weights' are deprecated sinc
nd may be removed in the future. The current behavior is equivalent to passing `weig
`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.7719
```

```
(base) gman@Grants-Air src % python f_1.py de-pgd_attack_light

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:208:
ning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
Open file in editor (cmd + click)
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:223:
ning: Arguments other than a weight enum or `None` for 'weights' are deprecated sinc
nd may be removed in the future. The current behavior is equivalent to passing `weig
`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.9115
```

```
(base) gman@Grants-MacBook-Air src % python f_1.py de-gaussian_blur

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:208:
ning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
please use 'weights' instead.
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:223:
ning: Arguments other than a weight enum or `None` for 'weights' are deprecated sinc
nd may be removed in the future. The current behavior is equivalent to passing `weig
`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.8998
```

```
(base) gman@Grants-Air src % python f_1.py baseline_original_images_color_avg

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:208
ning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
please use 'weights' instead.
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:223
ning: Arguments other than a weight enum or `None` for 'weights' are deprecated sinc
nd may be removed in the future. The current behavior is equivalent to passing `weig
`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.9708
```

```
(base) gman@Grants-Air src % python f_1.py de-pgd_attack_light_color_avg

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:208:
ning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
please use 'weights' instead.
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:223:
ning: Arguments other than a weight enum or `None` for 'weights' are deprecated sinc
nd may be removed in the future. The current behavior is equivalent to passing `weig
`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.8178
```

```
(base) gman@Grants-Air src % python f_1.py de-pgd_attack_light_color_avg

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:208:
ning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
please use 'weights' instead.
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:223:
ning: Arguments other than a weight enum or `None` for 'weights' are deprecated sinc
nd may be removed in the future. The current behavior is equivalent to passing `weig
`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.8178
```

```
(base) gman@Grants-Air src % python f_1.py baseline_pgd_attack

[INFO]: Not loading pre-trained weights
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:208:
ning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the
please use 'weights' instead.
  warnings.warn(
/Users/gman/anaconda3/lib/python3.11/site-packages/torchvision/models/_utils.py:223:
ning: Arguments other than a weight enum or `None` for 'weights' are deprecated sinc
nd may be removed in the future. The current behavior is equivalent to passing `weig
`.
  warnings.warn(msg)
[INFO]: Freezing hidden layers...
F1 Score: 0.8431
```

Solution Space (MobileNetV3)

Looking at only stop signs (113 images):

	Test Accuracy	F1 Score
Baseline with FGSM Attack	97.345%	0.986
With Canny Edge and dimensionality reduction	97.345%	0.986