Automated Stop Sign Tampering and Adversarial Attack Detection

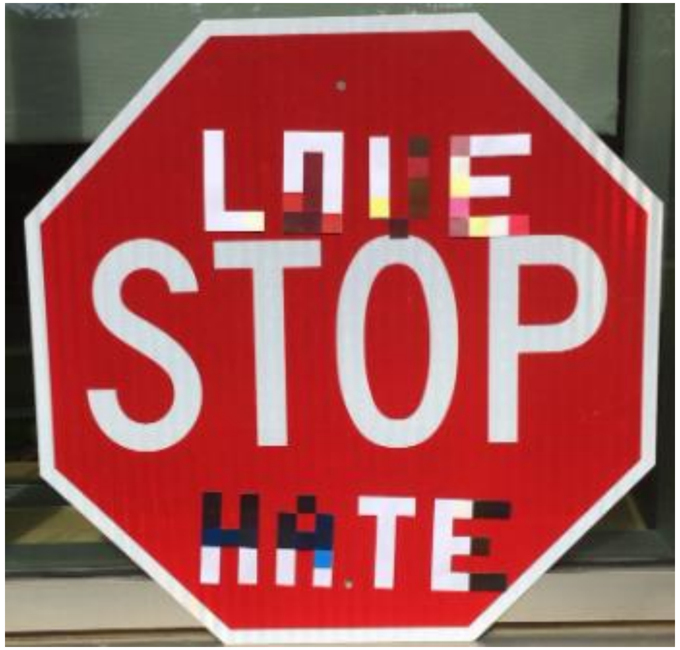
Jed Pulley, Rishika Ahuja, & Devin Bresser

Adversarial Attacks

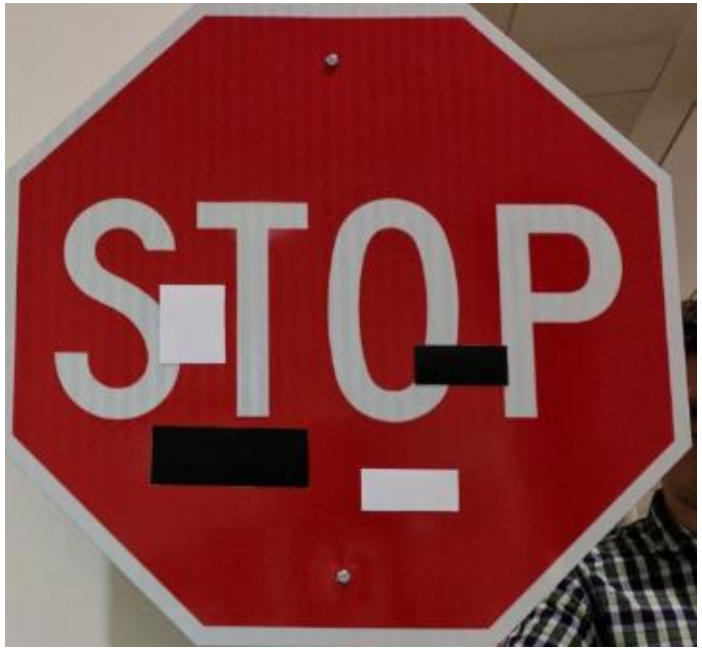
- Stop sign classification must be 100% foolproof
- Researchers at the University of Washington developed a method for altering stop signs that caused high percentages of misclassification
- They first trained a model, then developed an algorithm to create alterations undetectable to the human eye, but significant enough to drastically alter the classification results
- This is a textbook example of an Adversarial Attack

Speaker: Jed

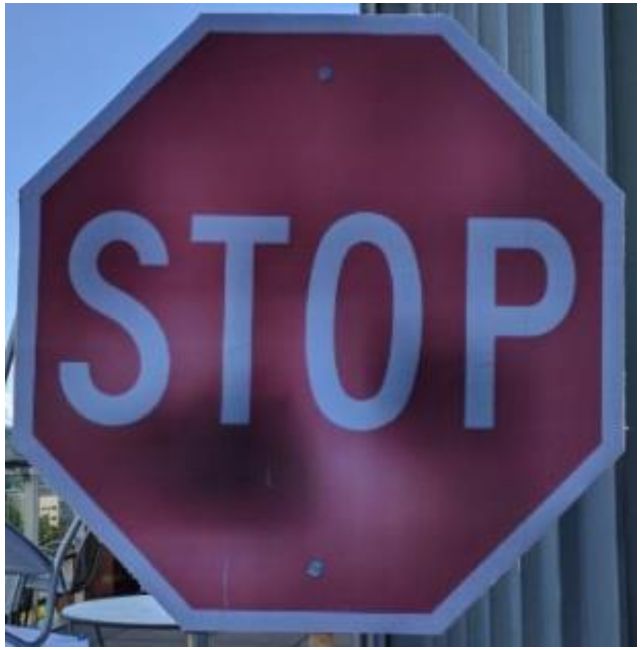
Motivating Example



Misclassified 2 of every 3 times [1]



Misclassified as 45mph speed limit sign [1]



Misclassified as 45mph sign 100% of the time [1]

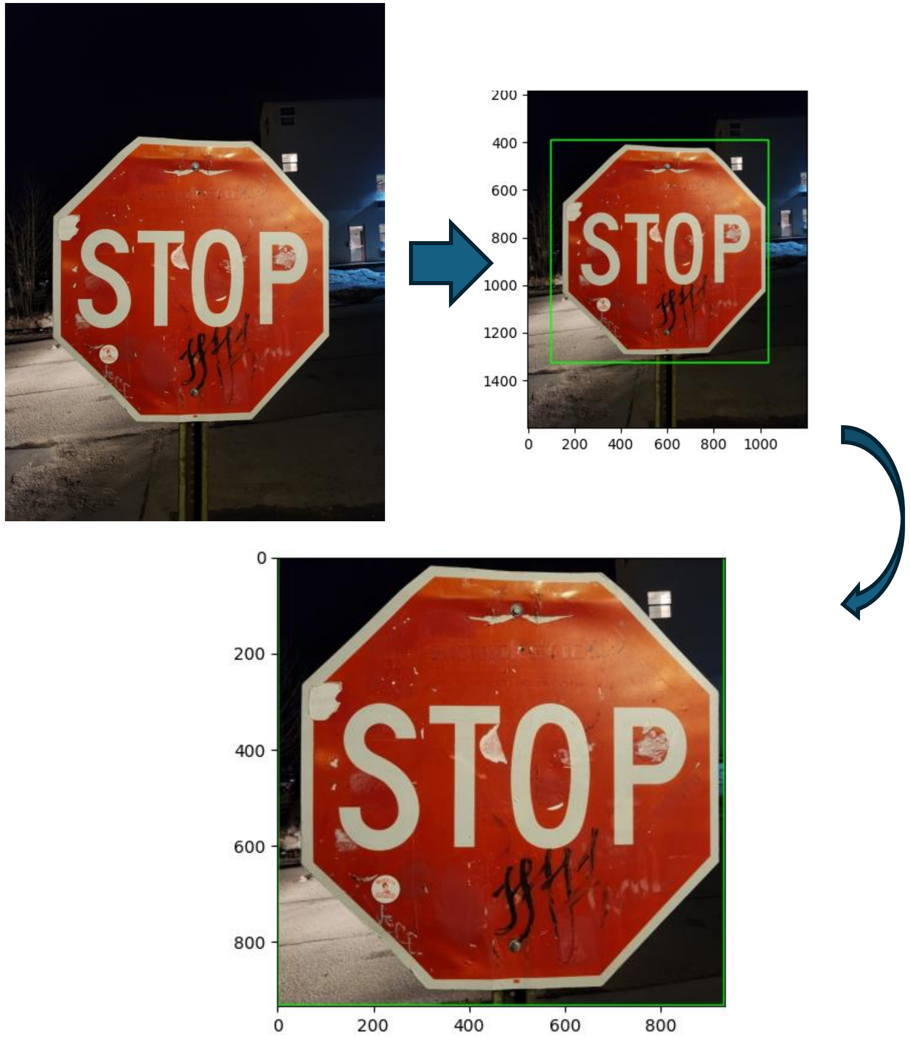
Speaker: Jed

[1] ArsTechnica

Methodology (1)

Bounding Box Creation and Cropping

- OpenCV was used to search for Stop-Signs using a pre-trained Cascade Classifier.
- A cascade classifier is a machine learning-based approach that is used for object detection in images.
- It works by using a cascade of multiple stages of classifiers to gradually eliminate areas of an image that are not likely to contain the object of interest, such as a face or a stop sign.
- This approach allows for fast and efficient object detection, making it suitable for real-time applications.
- If a Stop-Sign is found, a green bounding box is drawn around it.
- The image is then cropped around the green bounding box then the background is removed using salient object detection.
- Finally, the processed image is passed on to the CNN.



Speaker: Rishika

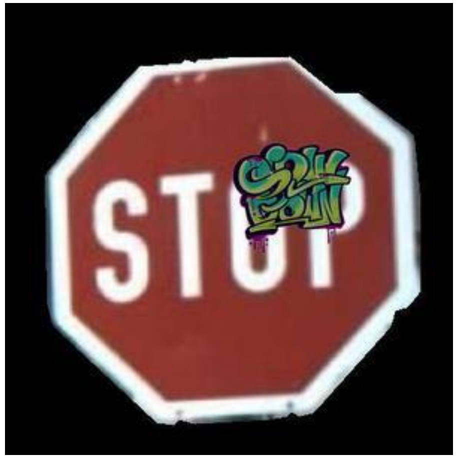
Methodology (2)

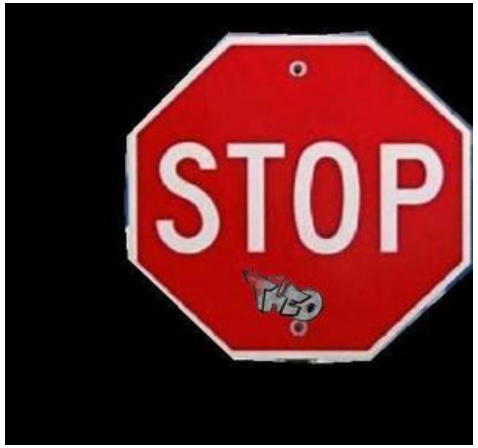
Training Data Synthesis

- Acquired and preprocessed ~120 stop sign images (with previous method)

Then, using Mathematica:

- Acquired and preprocessed 85 graffiti stock images
- Procedurally place graffiti onto stop signs in various orientations
- Procedurally alter output images (rotation, blur, sharpen, contrast)
- This gives a robust set of training data!
  - ~19,000 negatives, ~64,000 positives



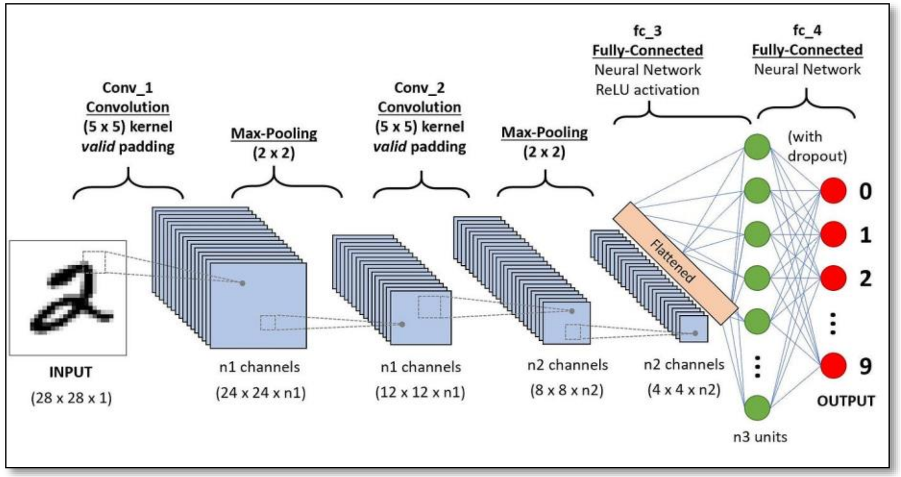


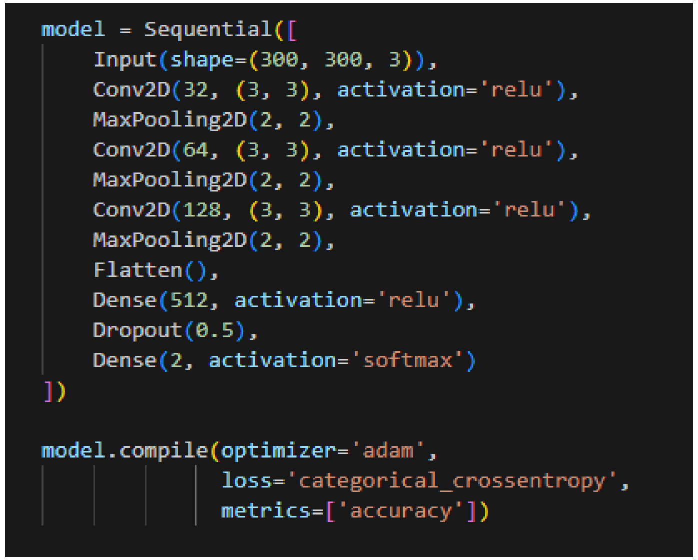
Speaker: Devin

Methodology (3)

Defining and Training a Convolutional Neural Network

- Now, problem becomes a binary classification problem on images.
- Can a computer distinguish between the graffitied and non-graffitied stop signs?
- Convolutional Neural Net (CNN): a deep learning model that excels at image classification.
    - Implement with Python & TensorFlow
- Simple architecture with three convolutional layers.
- Train on a small subset (1/10th) of randomly sampled synthetic data.





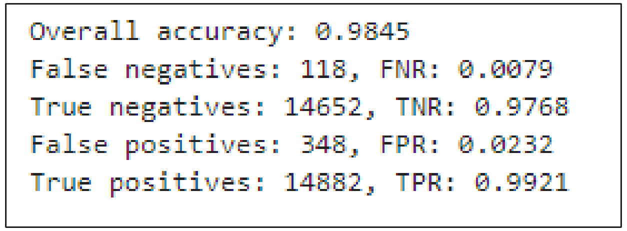
```
model = Sequential([
    Input(shape=(300, 300, 3)),
    Conv2D(32, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Flatten(),
    Dense(512, activation='relu'),
    Dropout(0.5),
    Dense(2, activation='softmax')
])

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Speaker: Devin

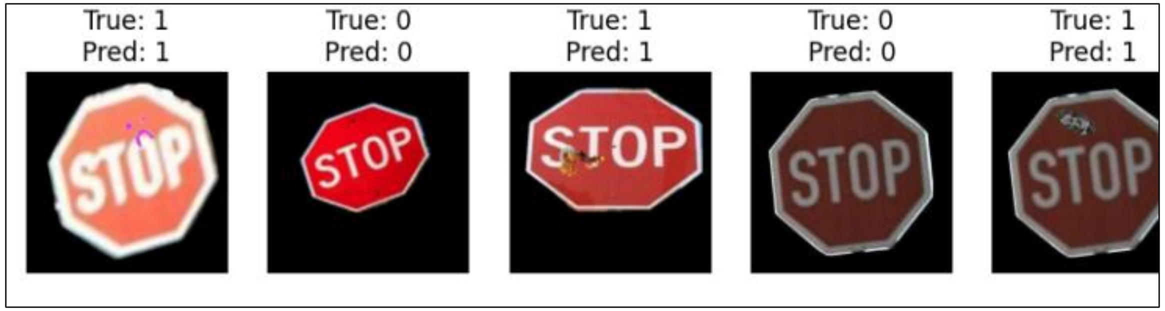
Results

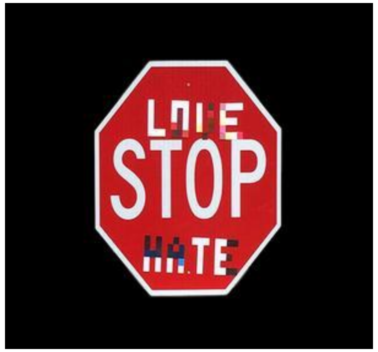
Model Results & Analysis

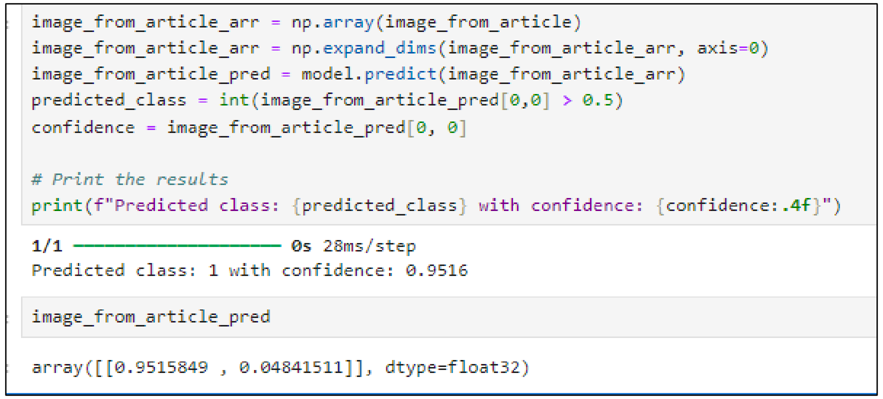
- "Small model" trained on 1/10th of the training data (~1,500 from each class) performs excellently on 30,000 test images.
- 98.5% overall accuracy, >99% true positive rate.
    - Model rarely ever predicts a normal stop sign as graffitied.
    - CNN exhibits very strong capability to generalize.
- Also tested on the motivating example for fun:
    - Model predicts 1 (graffitied) with 95% confidence.



```
Overall accuracy: 0.9845
False negatives: 118, FNR: 0.0079
True negatives: 14652, TNR: 0.9768
False positives: 348, FPR: 0.0232
True positives: 14882, TPR: 0.9921
```







```
image_from_article_arr = np.array(image_from_article)
image_from_article_arr = np.expand_dims(image_from_article_arr, axis=0)
image_from_article_pred = model.predict(image_from_article_arr)
predicted_class = int(image_from_article_pred[0,0] > 0.5)
confidence = image_from_article_pred[0, 0]

# Print the results
print(f"Predicted class: {predicted_class} with confidence: {confidence:.4f}")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Predicted class: 1 with confidence: 0.9516
```

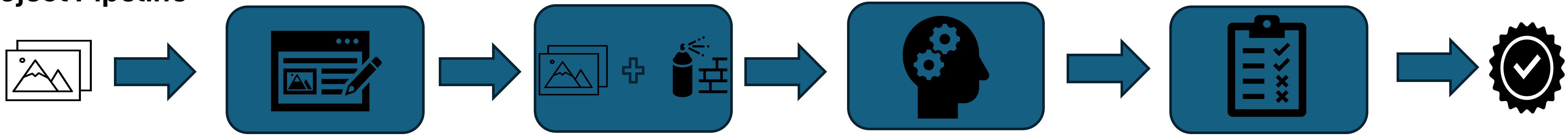
```
image_from_article_pred
```

```
array([[0.9515849 , 0.04841511]], dtype=float32)
```

Speaker: Devin

Conclusion

Project Pipeline



Created a full pipeline from raw images to insight as to whether a stop sign is graffiti'd or not.

- **Stop Sign Geometry Recognition and Cropping**: Use a computer vision algorithm to detect and crop stop signs in raw images.
- **Synthetic Training Data Generation**: Generate synthetic training data by applying various types of graffiti to the cropped stop sign images, creating a robust training dataset
- **CNN-based Binary Classification**: Train a Convolutional Neural Network (CNN) on the synthetic dataset to classify stop signs as either graffiti'd or not.
- **Evaluation and Performance**: Evaluate the trained CNN on a separate test dataset to assess its performance in detecting graffiti on stop signs- 98.5% overall accuracy, >99% true positive rate.
- **Insight Generation**: Once the model is trained and evaluated, use it to predict whether new stop sign images contain graffiti or not. The system can have practical applications in urban management, traffic safety, law enforcement, and the development of autonomous vehicle technology.

Speaker: Rishika

Any Questions?

References

- Hacking street signs with stickers could confuse self-driving cars - Jonathan M. Gitlin https://arstechnica.com/cars/2017/09/hacking-street-signs-with-stickers-could-confuse-self-driving-cars/
- Adversarial Attacks and Defences for Convolutional Neural Networks – Joao Gomes https://medium.com/onfido-tech/adversarial-attacks-and-defences-for-convolutional-neural-networks-66915ece52e7