howest
university of applied sciences

AI & Security Project

Using adversarial attacks to cause misclassification

Group 1

Harman Singh
Sadiksha Sapkota

Project overview

- **Project 2.2**
  - **Part 1**: Adversarial Attacks on Image Classification Models
    - Investigate the vulnerability of image classification models to adversarial attacks
    - Use pre-trained image classifiers
    - Explore adversarial attack methods
  - **Part 2**: Defense Mechanisms against Adversarial Attacks
    - Explore and investigate defense techniques that enhance robustness

Implementation

- **Part 1: Attack**
  - **Dataset**: TinyImageNet, Pretrained patches
  - **Attacks**:
    - Method 1: Adversarial attacks (FGSM, PGD, C&W)
    - Method 2: Adversarial patches (SGD)
  - **Pre-trained models**:
    - Method 1: ResNet18, ResNet50, ResNet152, VGG16, VGG19
    - Method 2: ResNet34, DensNet12
- **Part 2: Defense**
  - **Mechanism**: Adversarial training

Workflow of method 1 – Adversarial attacks

- **Evaluation order**
  1) baseline performance (without attack)
  2) attack performance (with all attacks)
  3) defense performance (with defense against all attacks)
- **Metrics**
  - **Top-1 error**: the number of times the correct class was not the predicted class
  - **Top-5 error**: the number of times the correct class was not in the top 5 predicted classes by certainty

howest university of applied sciences

Workflow of method 2 – Adversarial patches

Terminology: Adversarial patches are small, specially made images or patterns designed to trick AI models into making incorrect predictions.

- Here we have 5 pretrained patches so that we can fool the network into the desired label.
- The pretrained patches include toaster, goldfish, school bus, lipstick, pineapple
- We have for each patches 32 pixels, 48 pixels, 64 pixels.

Workflow of method 2 – Adversarial patches

Accuracy: Top-1, Top-5

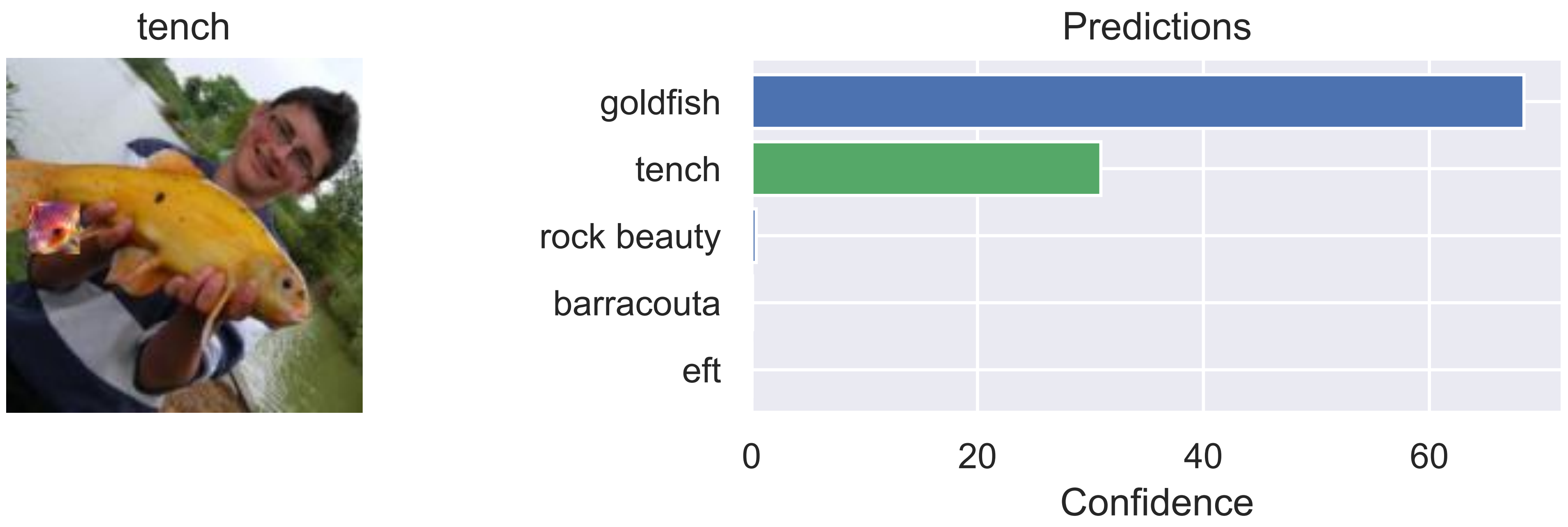
```
show_table(top_1=True)
```

Class name	Patch size 32x32	Patch size 48x48	Patch size 64x64
toaster	48.89%	90.48%	98.58%
goldfish	69.53%	93.53%	98.34%
school bus	78.79%	93.95%	98.22%
lipstick	43.36%	86.05%	96.41%
pineapple	79.74%	94.48%	98.72%

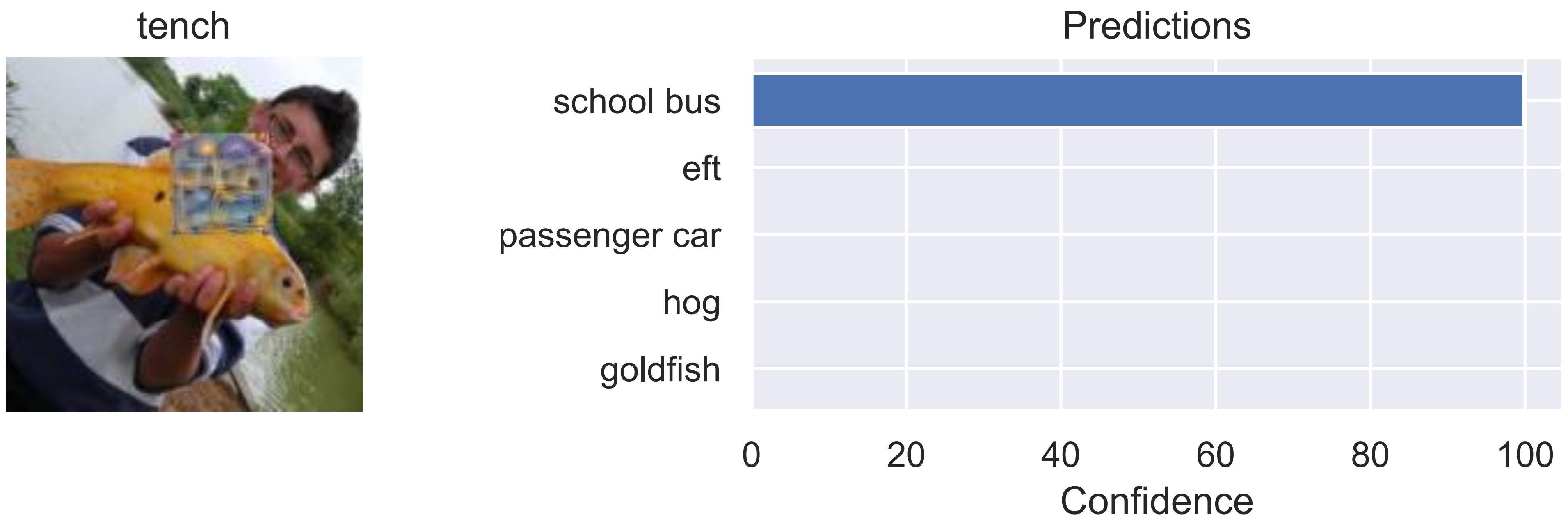
```
show_table(top_1=False)
```

Class name	Patch size 32x32	Patch size 48x48	Patch size 64x64
toaster	72.02%	98.12%	99.93%
goldfish	86.31%	99.07%	99.95%
school bus	91.64%	99.15%	99.89%
lipstick	70.10%	96.86%	99.73%
pineapple	92.23%	99.26%	99.96%

Workflow of method 2 – Adversarial patches




```
perform_patch_attack(patch_dict['goldfish'][32]['patch'])
```




```
perform_patch_attack(patch_dict['school bus'][64]['patch'])
```


howest university of applied sciences

Workflow of method 2 – Adversarial patches

Transferability

```python
transfer_model = torchvision.models.densenet121(weights='IMAGENET1K_V1')
transfer_model = transfer_model.to(device)

# No gradients needed for the network
transfer_model.eval()
for p in transfer_model.parameters():
    p.requires_grad = False
```

```python
class_name = 'pineapple'
patch_size = 64
print(f"Testing patch \"{class_name}\" of size {patch_size}x{patch_size}")

results = eval_patch(transfer_model,
                     patch_dict[class_name][patch_size]["patch"],
                     data_loader,
                     target_class=label_names.index(class_name))

print(f"Top-1 fool accuracy: {(results[0] * 100.0):4.2f}%")
print(f"Top-5 fool accuracy: {(results[1] * 100.0):4.2f}%")
```

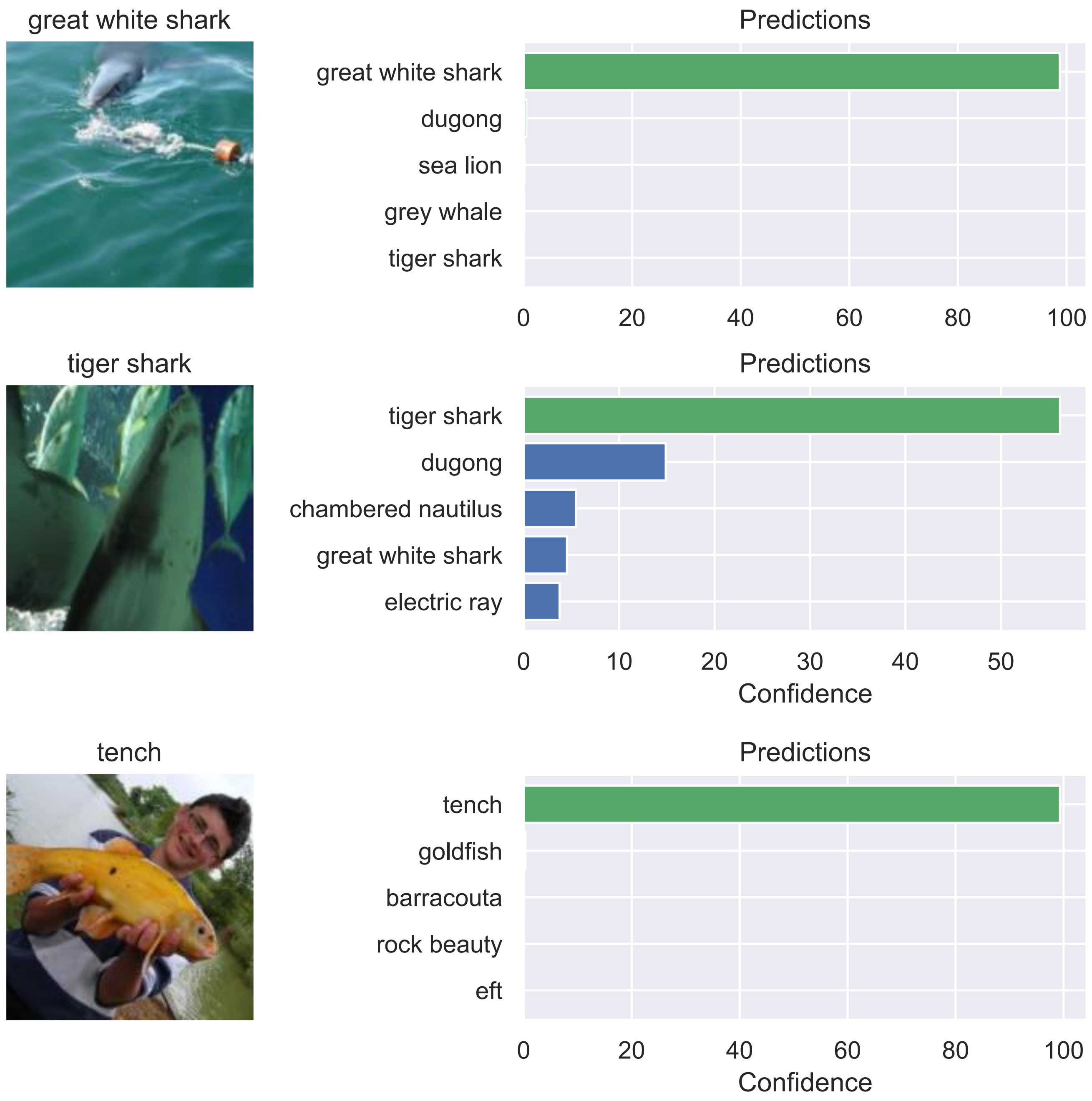
```
Testing patch "pineapple" of size 64x64

Validating...:   0%|          | 0/157 [00:00<?, ?it/s]

Top-1 fool accuracy: 64.89%
Top-5 fool accuracy: 82.21%
```

Our findings

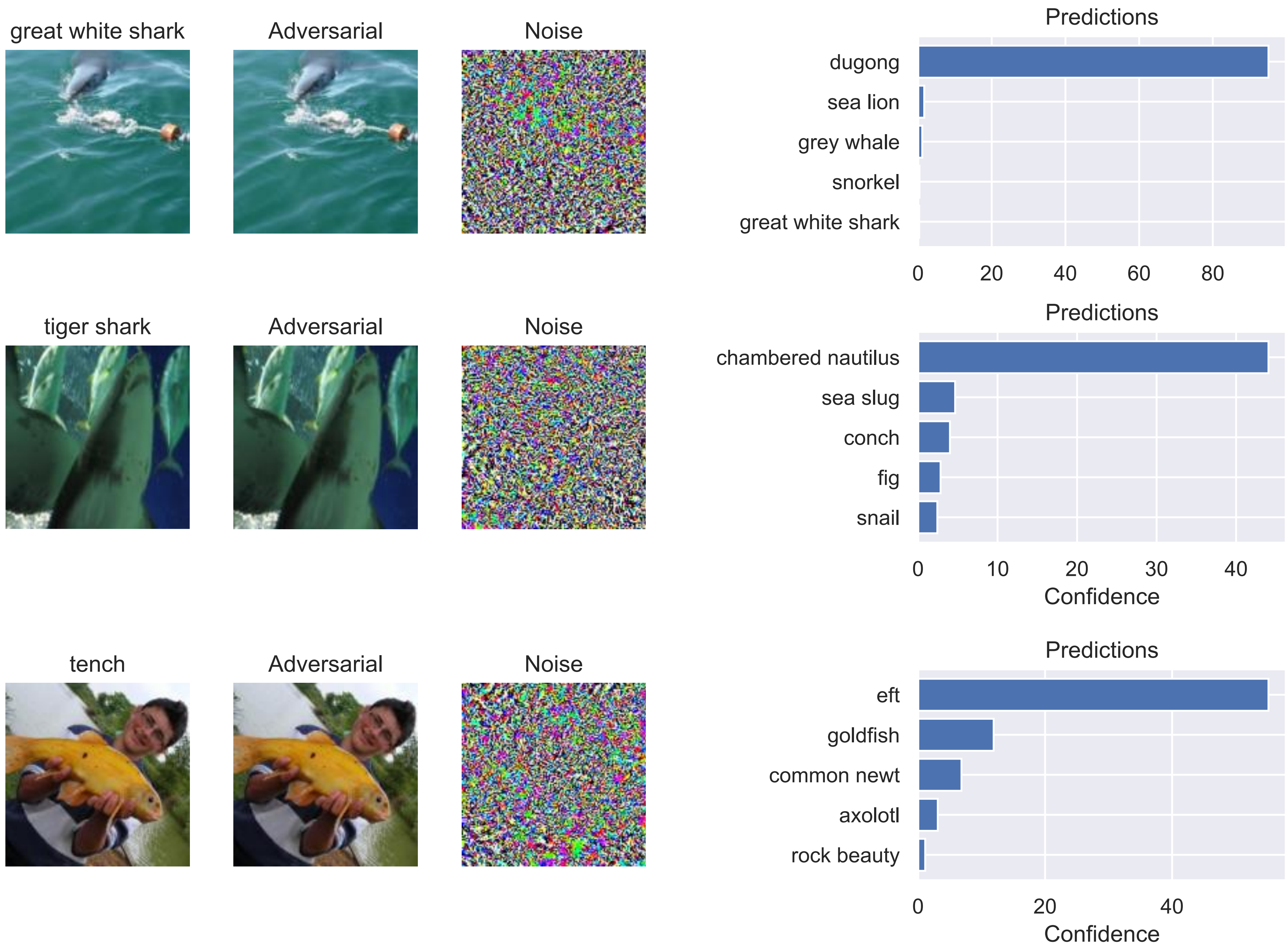
- **Attacking the models decreases model confidence, and increases likelihood of spreading predictions across multiple classes instead of just picking 1 class**
- The bigger the epsilon, the bigger the error
- PGD was the most effective attack method (up to 500% increase in Top-1 error rate)
  - Adversarial training against PGD did not seem to improve the model's performance as much
- For ResNet18 and ResNet50, adversarial training made the model perform better after it had been attack, than the baseline performance
- Adversarial training did not seem to have an impact on VGG16 nor VGG19

Our findings

Baseline



Attacked with FGSM



Our findings

- Attacking the models decreases model confidence, and increases likelihood of spreading predictions across multiple classes instead of just picking 1 class
- **The bigger the epsilon, the bigger the error**
- PGD was the most effective attack method (up to 500% increase in Top-1 error rate)
    - Adversarial training against PGD did not seem to improve the model’s performance as much
- For ResNet18 and ResNet50, adversarial training made the model perform better after it had been attack, than the baseline performance
- Adversarial training did not seem to have an impact on VGG16 nor VGG19

howest
university of applied sciences

Our findings

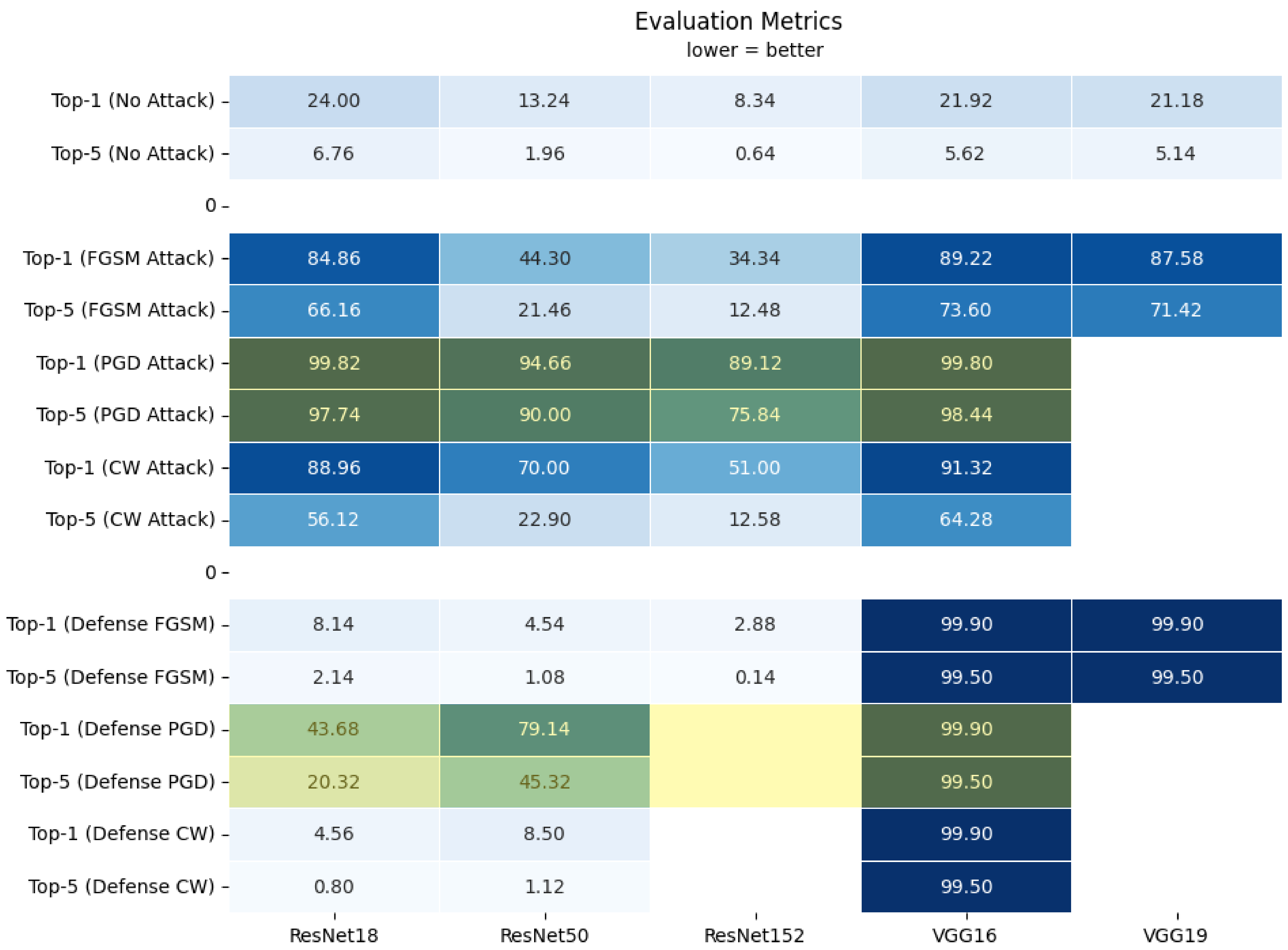
The bigger the epsilon, the bigger the error finding precise number is not always easy

```
Evaluating ResNet18 (FGSM) with epsilon 0.01:

        Top-1 error: 79.18%
        Top-5 error: 58.74%

Evaluating ResNet18 (FGSM) with epsilon 0.02:

        Top-1 error: 82.66%
        Top-5 error: 62.82%

Evaluating ResNet18 (FGSM) with epsilon 0.03:

        Top-1 error: 84.86%
        Top-5 error: 66.16%

Evaluating ResNet18 (FGSM) with epsilon 0.05:

        Top-1 error: 88.04%
        Top-5 error: 71.52%

Evaluating ResNet18 (FGSM) with epsilon 0.1:

        Top-1 error: 91.22%
        Top-5 error: 78.86%
```

Our findings

- Attacking the models decreases model confidence, and increases likelihood of spreading predictions across multiple classes instead of just picking 1 class
- The bigger the epsilon, the bigger the error
- **PGD was the most effective attack method**
  - **Adversarial training against PGD did not seem to improve the model's performance as much**
- For ResNet18 and ResNet50, adversarial training made the model perform better after it had been attack, than the baseline performance
- Adversarial training did not seem to have an impact on VGG16 nor VGG19

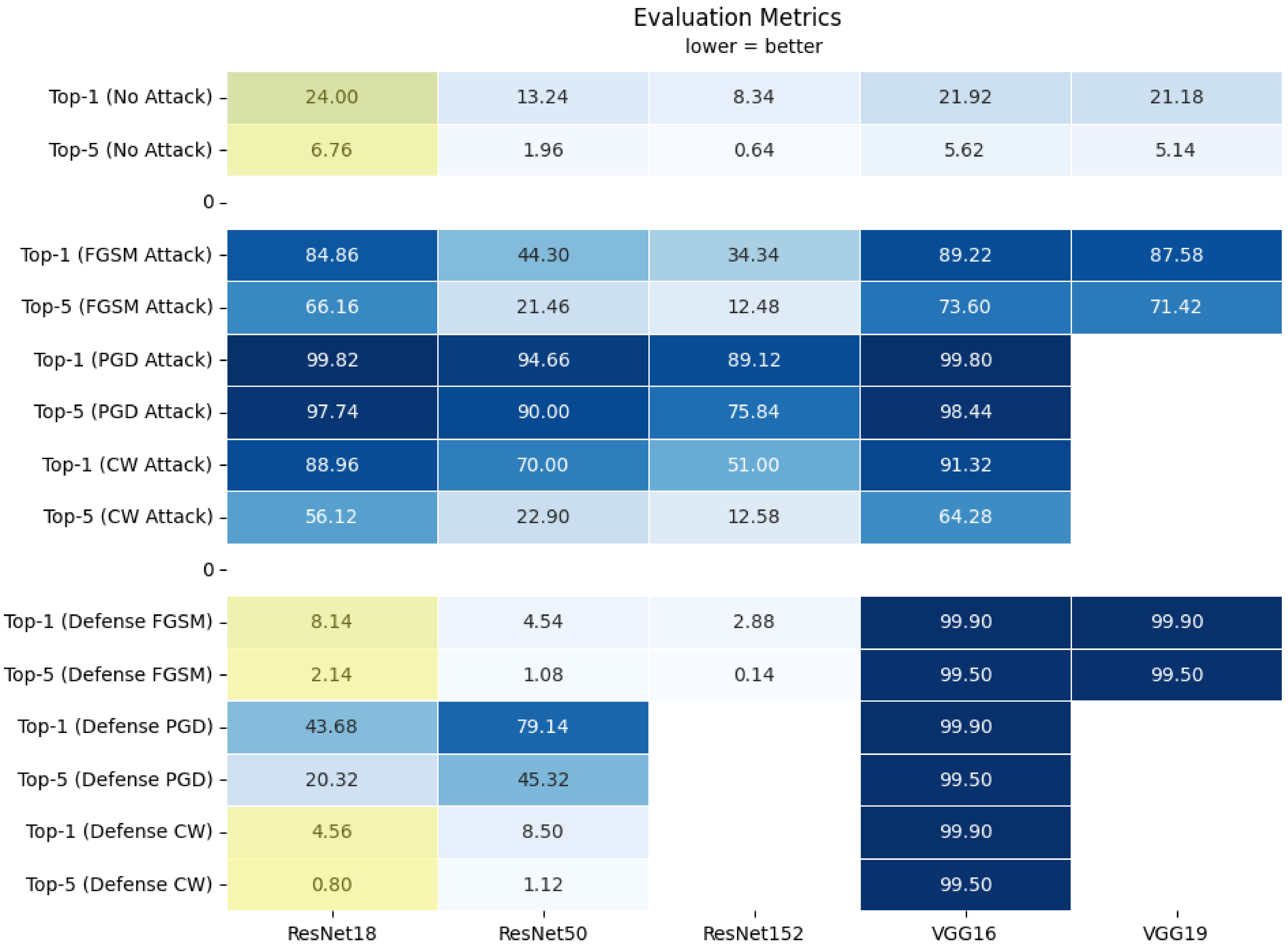
howest
university of applied sciences


Evaluation Metrics
lower = better
Top-1 (No Attack)
24.00
13.24
8.34
21.92
21.18
Top-5 (No Attack)
6.76
1.96
0.64
5.62
5.14
0
Top-1 (FGSM Attack)
84.86
44.30
34.34
89.22
87.58
Top-5 (FGSM Attack)
66.16
21.46
12.48
73.60
71.42
Top-1 (PGD Attack)
99.82
94.66
89.12
99.80
Top-5 (PGD Attack)
97.74
90.00
75.84
98.44
Top-1 (CW Attack)
88.96
70.00
51.00
91.32
Top-5 (CW Attack)
56.12
22.90
12.58
64.28
0
Top-1 (Defense FGSM)
8.14
4.54
2.88
99.90
99.90
Top-5 (Defense FGSM)
2.14
1.08
0.14
99.50
99.50
Top-1 (Defense PGD)
43.68
79.14
99.90
Top-5 (Defense PGD)
20.32
45.32
99.50
Top-1 (Defense CW)
4.56
8.50
99.90
Top-5 (Defense CW)
0.80
1.12
99.50
ResNet18
ResNet50
ResNet152
VGG16
VGG19

Our findings

- Attacking the models decreases model confidence, and increases likelihood of spreading predictions across multiple classes instead of just picking 1 class
- The bigger the epsilon, the bigger the error but finding percise.
- PGD was the most effective attack method (up to 500% increase in Top-1 error rate)
  - Adversarial training against PGD did not seem to improve the model's performance as much
- **For ResNet18 and ResNet50, adversarial training made the model perform better after it had been attack, than the baseline performance**
- Adversarial training did not seem to have an impact on VGG16 nor VGG19

howest
university of applied sciences


Evaluation Metrics
lower = better
Top-1 (No Attack) 24.00 13.24 8.34 21.92 21.18
Top-5 (No Attack) 6.76 1.96 0.64 5.62 5.14
0
Top-1 (FGSM Attack) 84.86 44.30 34.34 89.22 87.58
Top-5 (FGSM Attack) 66.16 21.46 12.48 73.60 71.42
Top-1 (PGD Attack) 99.82 94.66 89.12 99.80
Top-5 (PGD Attack) 97.74 90.00 75.84 98.44
Top-1 (CW Attack) 88.96 70.00 51.00 91.32
Top-5 (CW Attack) 56.12 22.90 12.58 64.28
0
Top-1 (Defense FGSM) 8.14 4.54 2.88 99.90 99.90
Top-5 (Defense FGSM) 2.14 1.08 0.14 99.50 99.50
Top-1 (Defense PGD) 43.68 79.14 99.90
Top-5 (Defense PGD) 20.32 45.32 99.50
Top-1 (Defense CW) 4.56 8.50 99.90
Top-5 (Defense CW) 0.80 1.12 99.50
ResNet18
ResNet50
ResNet152
VGG16
VGG19

Our findings

- Attacking the models decreases model confidence, and increases likelihood of spreading predictions across multiple classes instead of just picking 1 class
- The bigger the epsilon, the bigger the error but finding percise.
- PGD was the most effective attack method (up to 500% increase in Top-1 error rate)
    - Adversarial training against PGD did not seem to improve the model’s performance as much
- For ResNet18 and ResNet50, adversarial training made the model perform better after it had been attack, than the baseline performance
- **Adversarial training did not seem to have an impact on VGG16 nor VGG19**

Evaluation Metrics

lower = better

	ResNet18	ResNet50	ResNet152	VGG16	VGG19
Top-1 (No Attack)	24.00	13.24	8.34	21.92	21.18
Top-5 (No Attack)	6.76	1.96	0.64	5.62	5.14
0					
Top-1 (FGSM Attack)	84.86	44.30	34.34	89.22	87.58
Top-5 (FGSM Attack)	66.16	21.46	12.48	73.60	71.42
Top-1 (PGD Attack)	99.82	94.66	89.12	99.80	
Top-5 (PGD Attack)	97.74	90.00	75.84	98.44	
Top-1 (CW Attack)	88.96	70.00	51.00	91.32	
Top-5 (CW Attack)	56.12	22.90	12.58	64.28	
0					
Top-1 (Defense FGSM)	8.14	4.54	2.88	99.90	99.90
Top-5 (Defense FGSM)	2.14	1.08	0.14	99.50	99.50
Top-1 (Defense PGD)	43.68	79.14		99.90	
Top-5 (Defense PGD)	20.32	45.32		99.50	
Top-1 (Defense CW)	4.56	8.50		99.90	
Top-5 (Defense CW)	0.80	1.12		99.50	