Explaining Neural Models for Image Classification

Rishabh Ranjan – 2018CS10416

MTQ315 Presentation

Part I: Explanations

SHAP

SHapley Additive ExPlanations[1]

- Additive feature attribution:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \,;\; z' \in \{0,1\}^M \rightarrow \text{simplified inputs},\; g \rightarrow \text{local approximation}$$

- Shapley values:

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} f_x(z') - f_x(z' \setminus i)$$

[1] Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." *Advances in neural information processing systems* 30 (2017).

Approximations

Exact SHAP is Intractable

- **KernelSHAP** (Linear LIME[2] + Shapley values)
  - model-agnostic approximation
  - `KernelExplainer` class in `shap` (Python library)

- **DeepSHAP** (DeepLIFT[3] + Shapley values)
  - optimised for deep neural networks
  - `DeepExplainer` class in `shap` (Python library)

[2] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016.
[3] Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje. "Learning important features through propagating activation differences." *International conference on machine learning*. PMLR, 2017.

Task

Motivation

What is the model looking at?

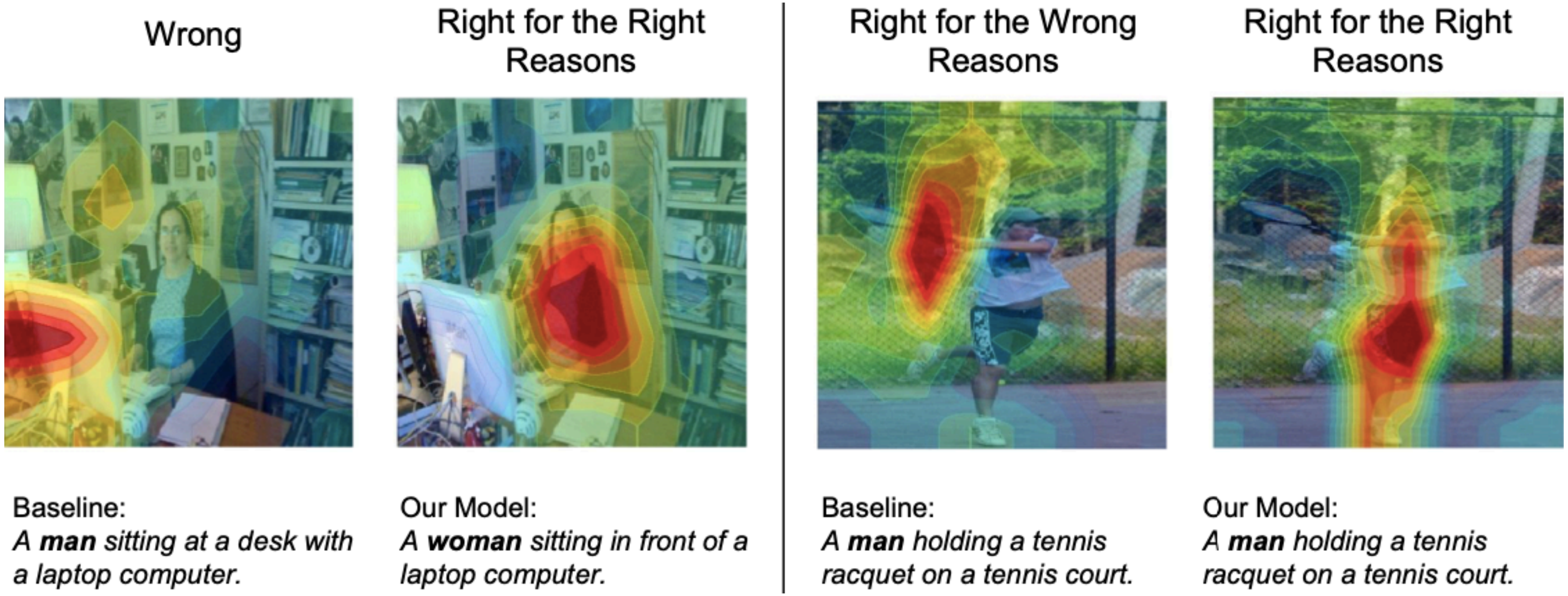


Figure from [4]. XAI methods applied to various models expose gender bias.

[4] Hendricks, Lisa Anne, et al. "Women also snowboard: Overcoming bias in captioning models." *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

Image Classification

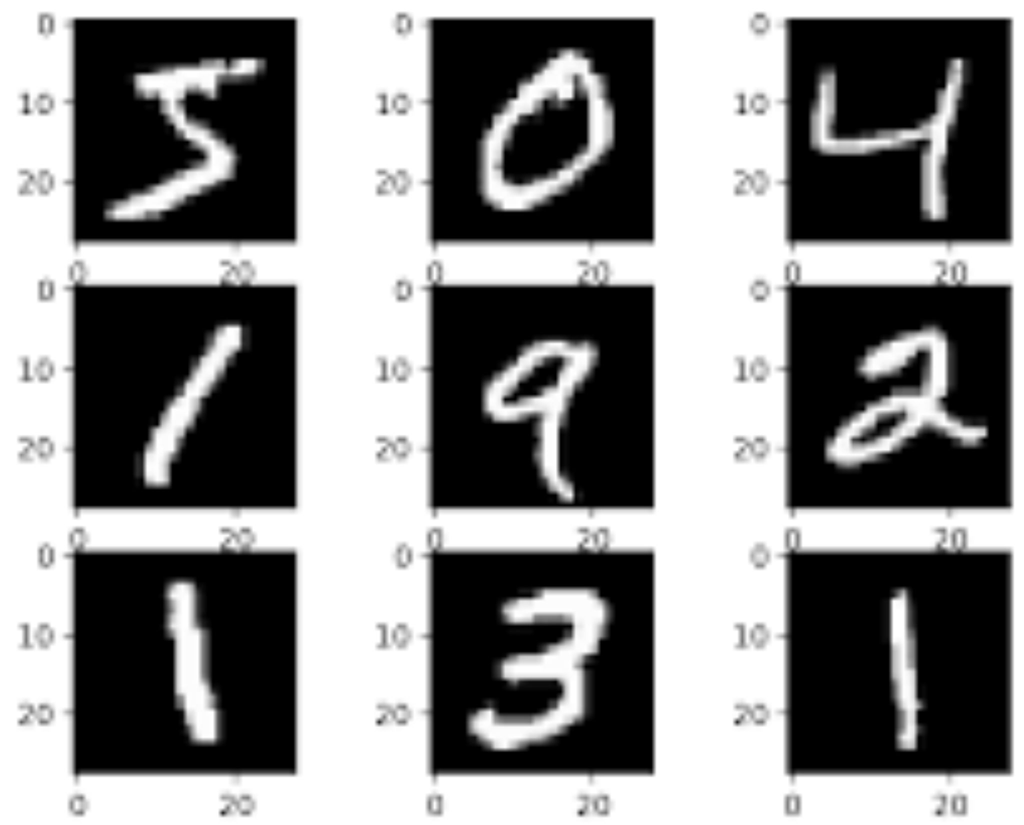
Multi-Class Classification on MNIST[5]-like Datasets

- **Input:** 28 x 28 grayscale image (pixel values 0 to 255)
- **Classes:** 0 to 9
- **Training examples:** 60,000
- **Testing examples:** 10,000

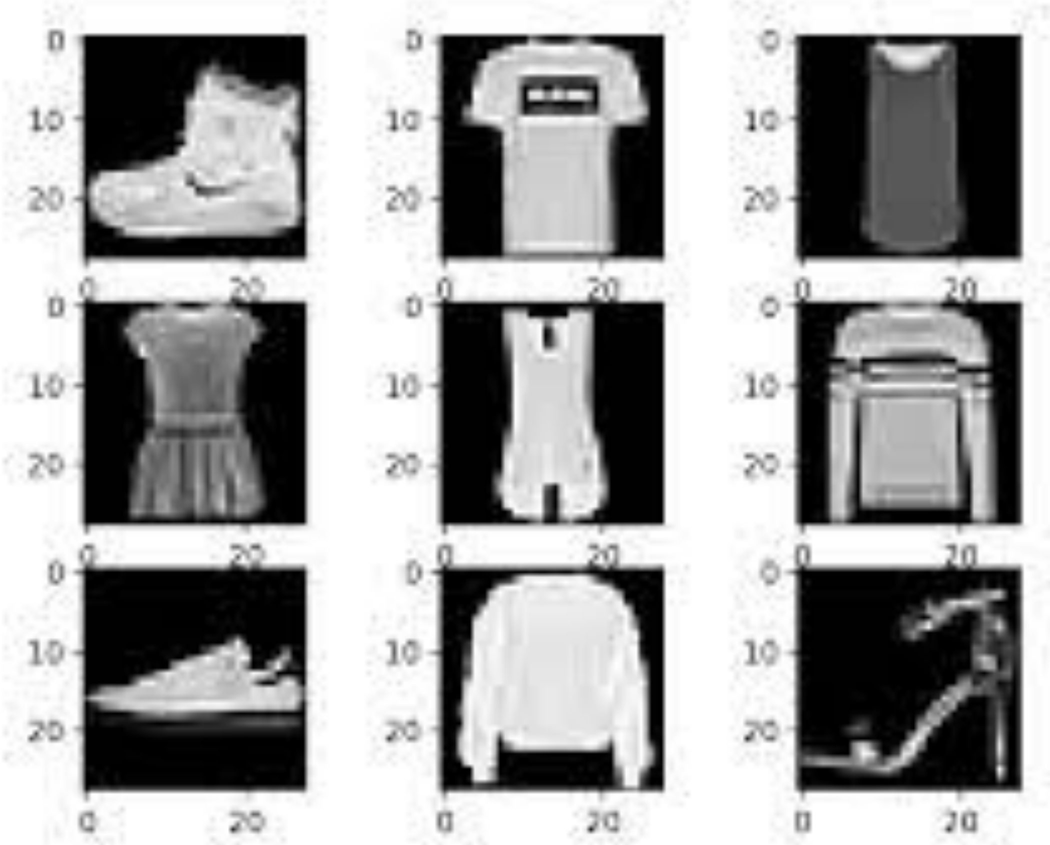
[5] Deng, Li. "The mnist database of handwritten digit images for machine learning research [best of the web]." *IEEE signal processing magazine* 29.6 (2012): 141-142.

Datasets

- **MNIST** (Modified National Institute of Standards and Technology)
  - handwritten digits
  - classes: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9



- **FMNIST** (Fashion MNIST)
  - items of clothing
  - classes: top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, boot



Results

Current Progress

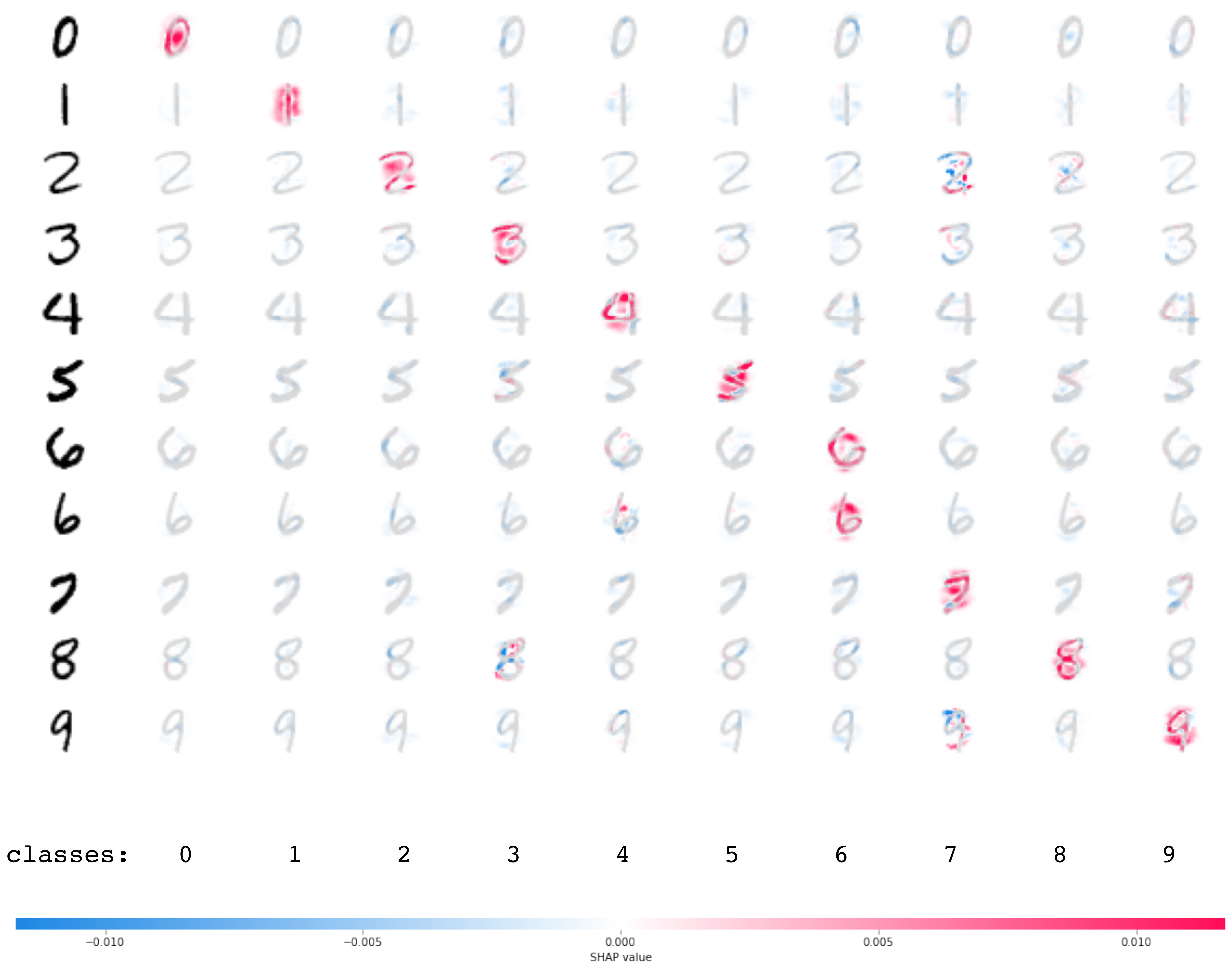
- **Datasets:** MNIST, FMNIST
- **Explainer:** DeepSHAP, ~~KernelSHAP~~
- **Model:**

```
self.conv_layers = nn.Sequential(
            nn.Conv2d(1, 10, kernel_size=5),
            nn.MaxPool2d(2),
            nn.ReLU(),
            nn.Conv2d(10, 20, kernel_size=5),
            nn.Dropout(),
            nn.MaxPool2d(2),
            nn.ReLU(),
        )
        self.fc_layers = nn.Sequential(
            nn.Linear(320, 50),
            nn.ReLU(),
            nn.Dropout(),
            nn.Linear(50, 10),
            nn.Softmax(dim=1)
        )
```

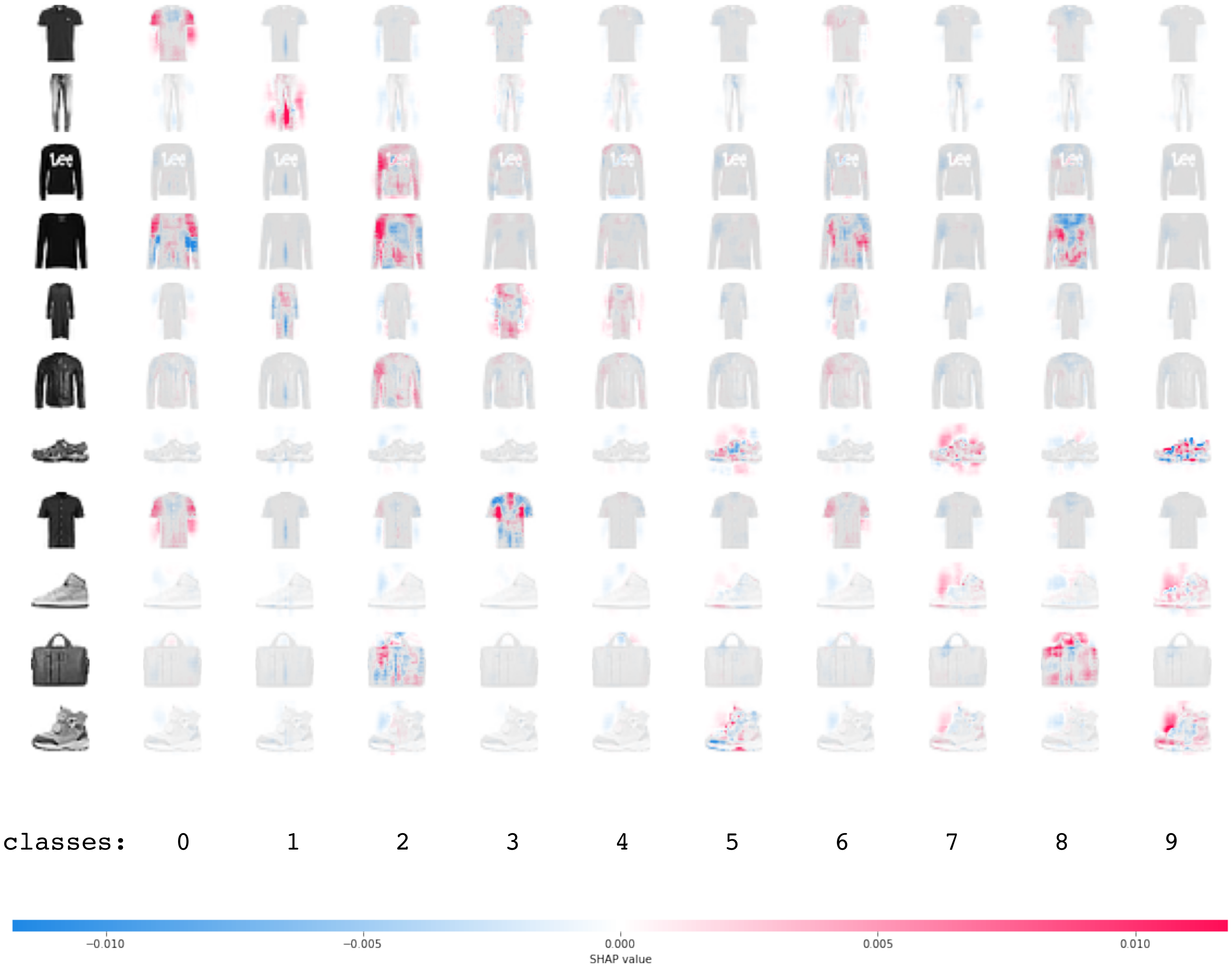
Interpretation

SHAP Visualization for Multi-Class Classification

- One-vs-Rest binary classification, per class
- BLUE: absence of pixels predicts class
- RED: presence of pixels predicts class
- But, `shap` library inverts colors in visualization:
  - MNIST has white foreground, and black background
- So, in visualization:
  - BLUE: absence of pixels predicts class
  - RED: presence of pixels predicts class
- Red blob on the correct class is due to local approximation



MNIST Visualisation (Acc. 99%)



	Label
0	Top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Boot

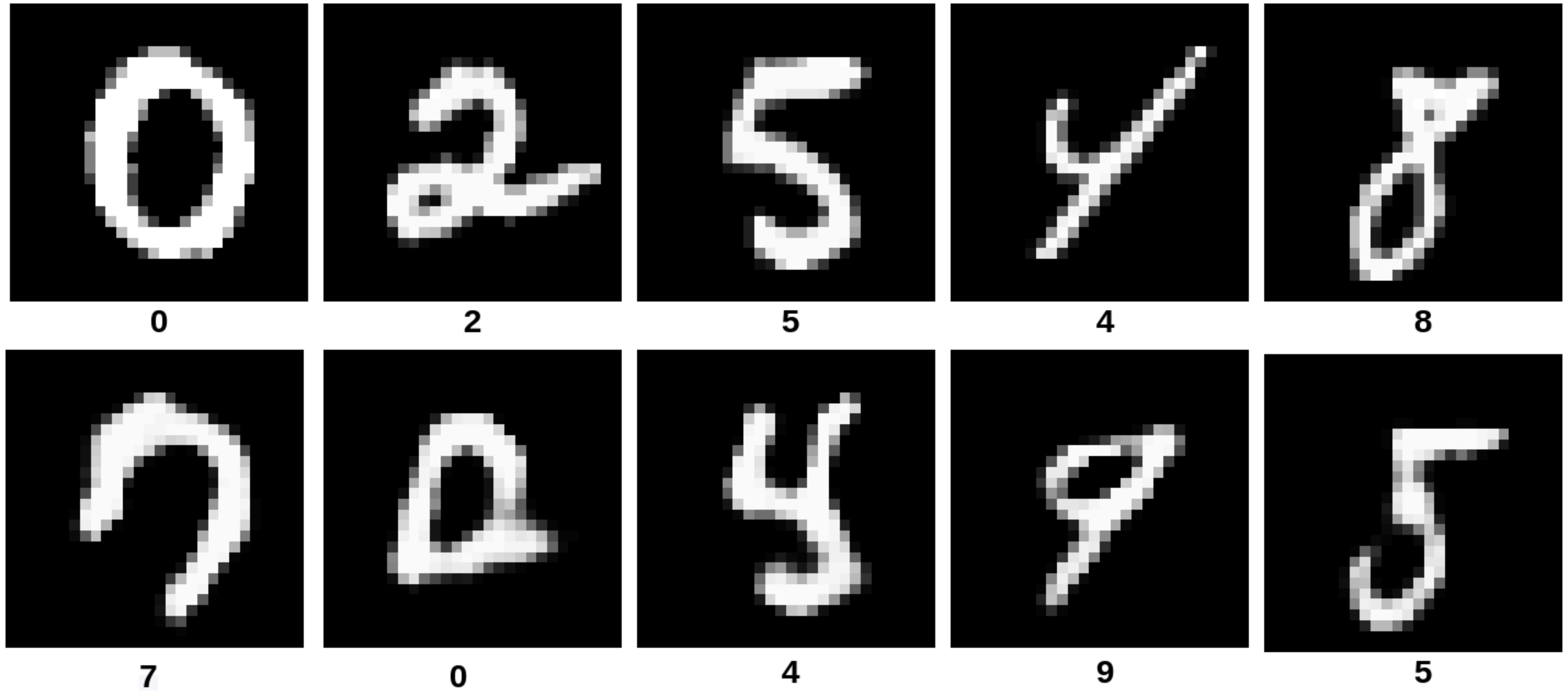
FMNIST Visualisation (Acc. 85%)

Part II: Counterfactuals

Motivation

Counterfactuals for Image Classification

- How to change pixels to obtain desired class?



TOP: Input and predicted class
BOTTOM: Counterfactual for desired class
(Figure from [6])

[6] Samoilescu, Robert-Florian, Arnaud Van Looveren, and Janis Klaise. "Model-agnostic and Scalable Counterfactual Explanations via Reinforcement Learning." *arXiv preprint arXiv:2106.02597* (2021).

Method 1: The First Approach[7]

- $\mathscr{L}(x' \,|\, x) = (f_t(x') - p_t)^2 + \lambda \|x' - x\|_1$, where,

  $x \rightarrow$ input

  $x' \rightarrow$ counterfactual

  $\mathscr{L} \rightarrow$ loss

  $f_t \rightarrow$ model prediction at class t

  $p_t \rightarrow$ desired probability of class t (typically $p_t = 1$)

  $\lambda \rightarrow$ hyperparameter

  $\|.\|_1 \rightarrow L_1$-norm (for **sparse** changes)

- `Counterfactual` class in alibi (Python library)

[7] Wachter, Sandra, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR." *Harv. JL & Tech.* 31 (2017): 841.

Method 2: Guided by Prototypes[8]

- Fast
- Model agnostic
- Interpretable counterfactuals (**in-distribution**)
- Uses class **prototypes**
- `CounterfactualProto` class in alibi (Python library)

[8] Looveren, Arnaud Van, and Janis Klaise. "Interpretable counterfactual explanations guided by prototypes." *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, Cham, 2021.

Method 3: Via Reinforcement Learning[9]

- Fast, model agnostic
- Does **not** assume model differentiability
- Allows flexible feature range constraints
  - eg. immutable protected features like *gender*, *race*, etc.
- RL technique: Deep Deterministic Policy Gradient (**DDPG**)
- `CounterfactualRL` class in alibi (Python library)

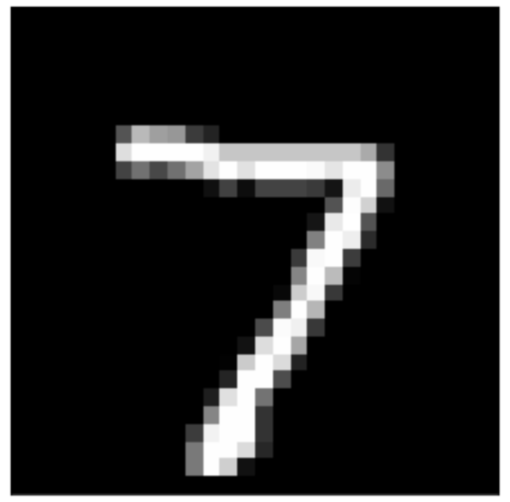
[9] Samoilescu, Robert-Florian, Arnaud Van Looveren, and Janis Klaise. "Model-agnostic and Scalable Counterfactual Explanations via Reinforcement Learning." *arXiv preprint arXiv:2106.02597* (2021).

Model details

- **Framework:** Tensorflow2 + Keras
- **MNIST test accuracy:** 98.6 %
- **FMNIST test accuracy:** 88.71 %

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 28, 28, 1)]       0
_________________________________________________________________
conv2d (Conv2D)              (None, 28, 28, 64)        320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 14, 14, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 14, 14, 32)        8224
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 7, 7, 32)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 7, 7, 32)          0
_________________________________________________________________
flatten (Flatten)            (None, 1568)              0
_________________________________________________________________
dense (Dense)                (None, 256)               401664
_________________________________________________________________
dropout_2 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                2570
=================================================================
Total params: 412,778
Trainable params: 412,778
Non-trainable params: 0
```

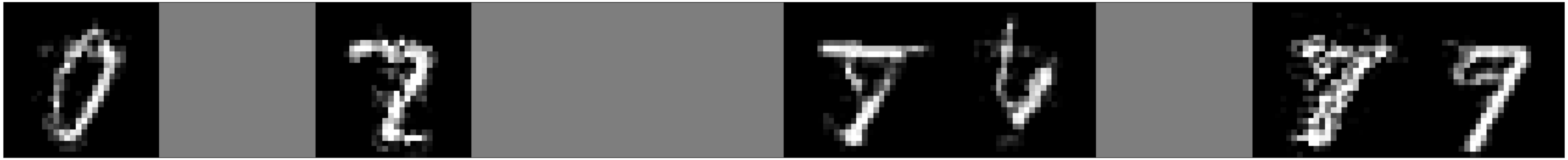
Results



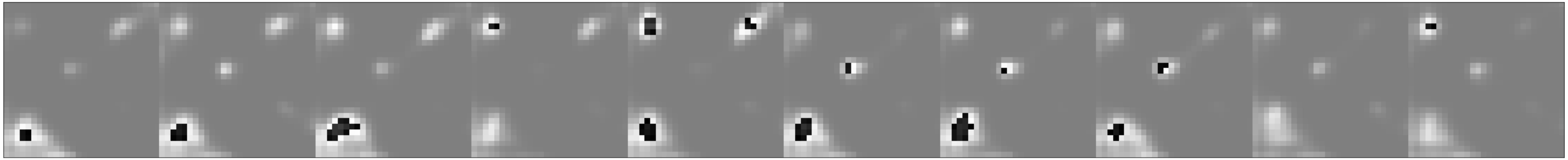
Input Image

- Gray $\implies$ counterfactual not found
- Method 3 failed to train due to bug in library

Method 1: Simple



Method 2: Prototype



Method 3: RL

Thank You!