RUB

RUHR-UNIVERSITÄT BOCHUM

Making the Training Visible

Computer Vision Deep Learning Project

Done by: Aya Altamimi, Simon Friz, Hyovin Kwak, Dimitri Tarnavski, Arslan Gabdulkhakov

Introduction

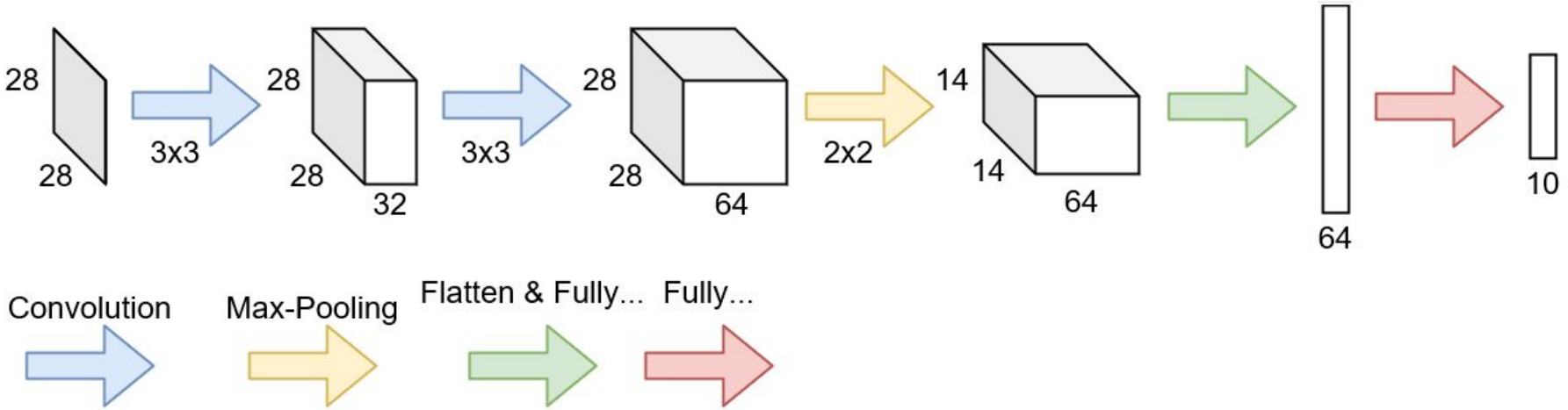
- Why DeConvnet
- What happens inside the network?
  - Sparse knowledge about deeper layers
- Go backwards through the network
  - Visualization technique based on Decovnet
- Deconvnet

Network architecture and training

Model

MNIST:60000 images, 6000 in each class
RPS Datasets 2520, 840 in each class

- ➔ **Input (28x28 or 300x300)**
- ➔ **Conv2D layer (32 filters 3x3)**
- ➔ **Conv2D layer (64 filters 3x3)**
- ➔ **Max pooling (2x2)**
- ➔ **Dense (64)**
- ➔ **Output (10 or 3)**

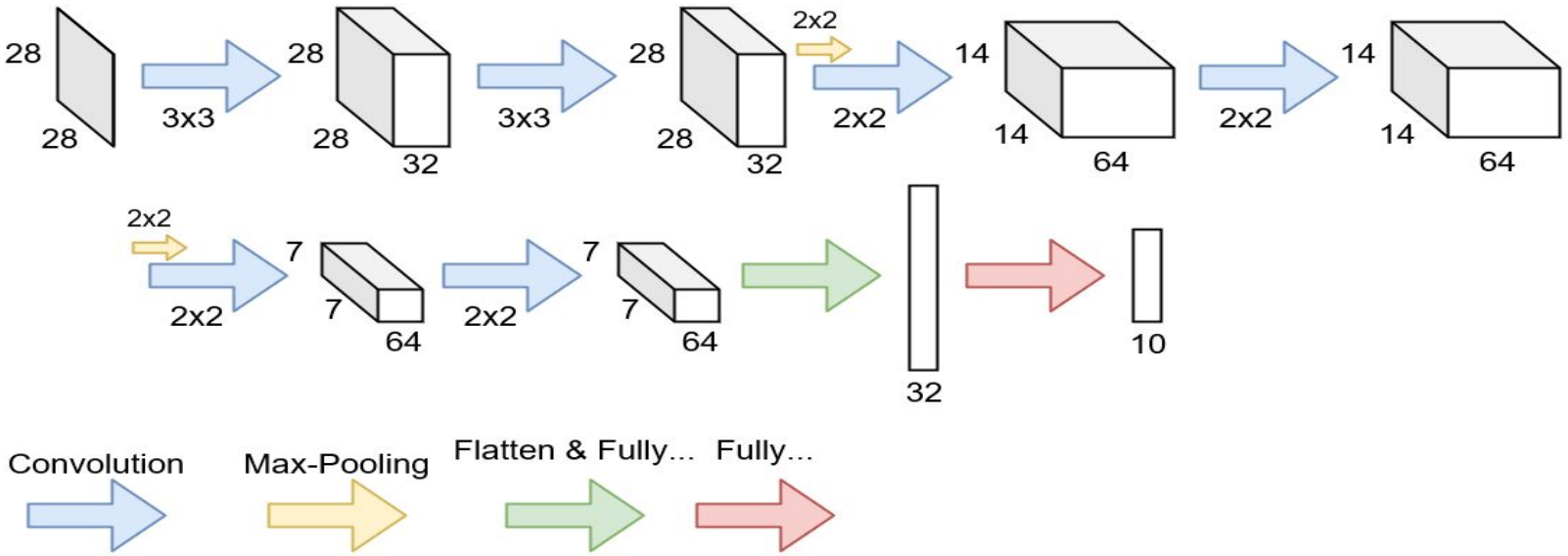


RUHR UNIVERSITÄT BOCHUM RUB

Network architecture and training

Deeper model

- ➔ Input (28x28 or 300x300)
- ➔ Conv2D layer (32 filters 3x3)
- ➔ Conv2D layer (64 filters 3x3)
- ➔ Max pooling (2x2)
- ➔ Dense (32)
- ➔ Output (10 or 3)



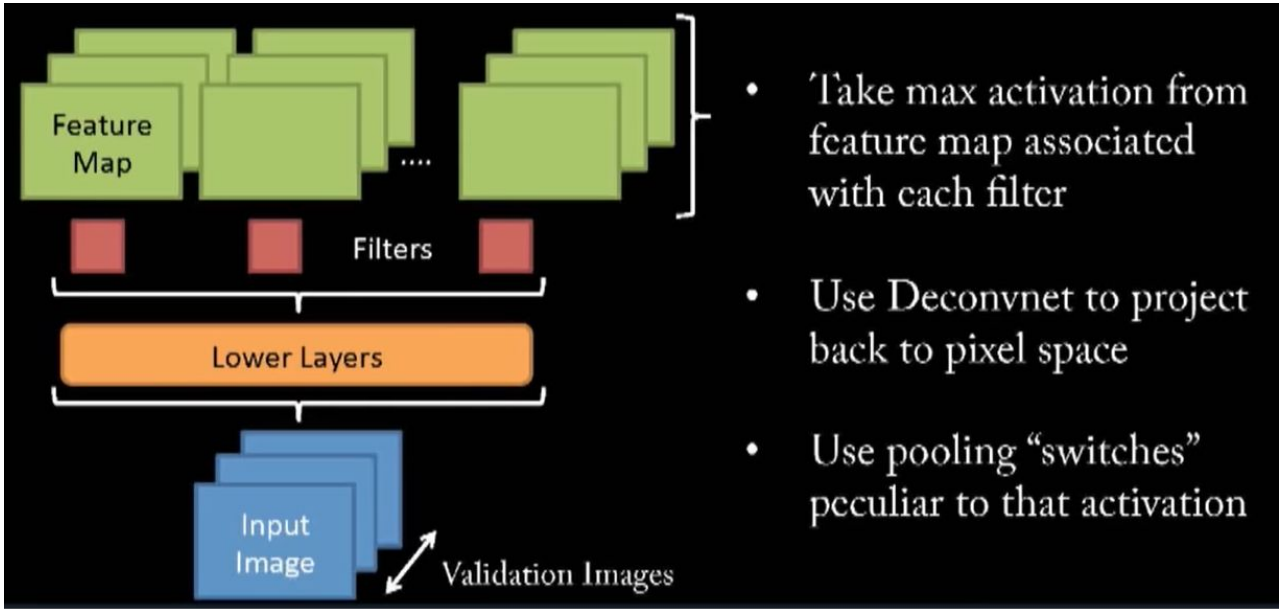
RUHR UNIVERSITÄT BOCHUM RUB

DeconvNet

- **Idea:**
  - **Mapping** activities **back** to the input **pixel space**.
    (Instead of mapping pixels to features, do it **the other way around**)
  - **Reconstruct** the activity in the **layer beneath** that gave rise to the chosen activation.
    (Reconstruct an approximate version of the convnet features from the layer beneath)
  - **Repeated** until input pixel space is reached.
    (DeconvNet **attached to each layer**)
  - **Not** used in any **learning** capacity. (Deconvnet is only used to project back into pixel space).

Projecting back from higher layers

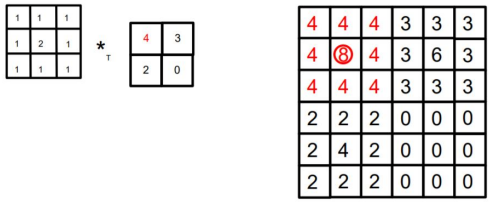
- **Run** input image through convolutional network **(Feed forward)**
- Results in **activations** over feature maps in **all the layers**
- Select a single feature map.
- Take strongest activation in that feature map of input image
  - Gives an idea what in the input image that activates the selected FM strongly
- Use that feature map as input of the DeconvNet

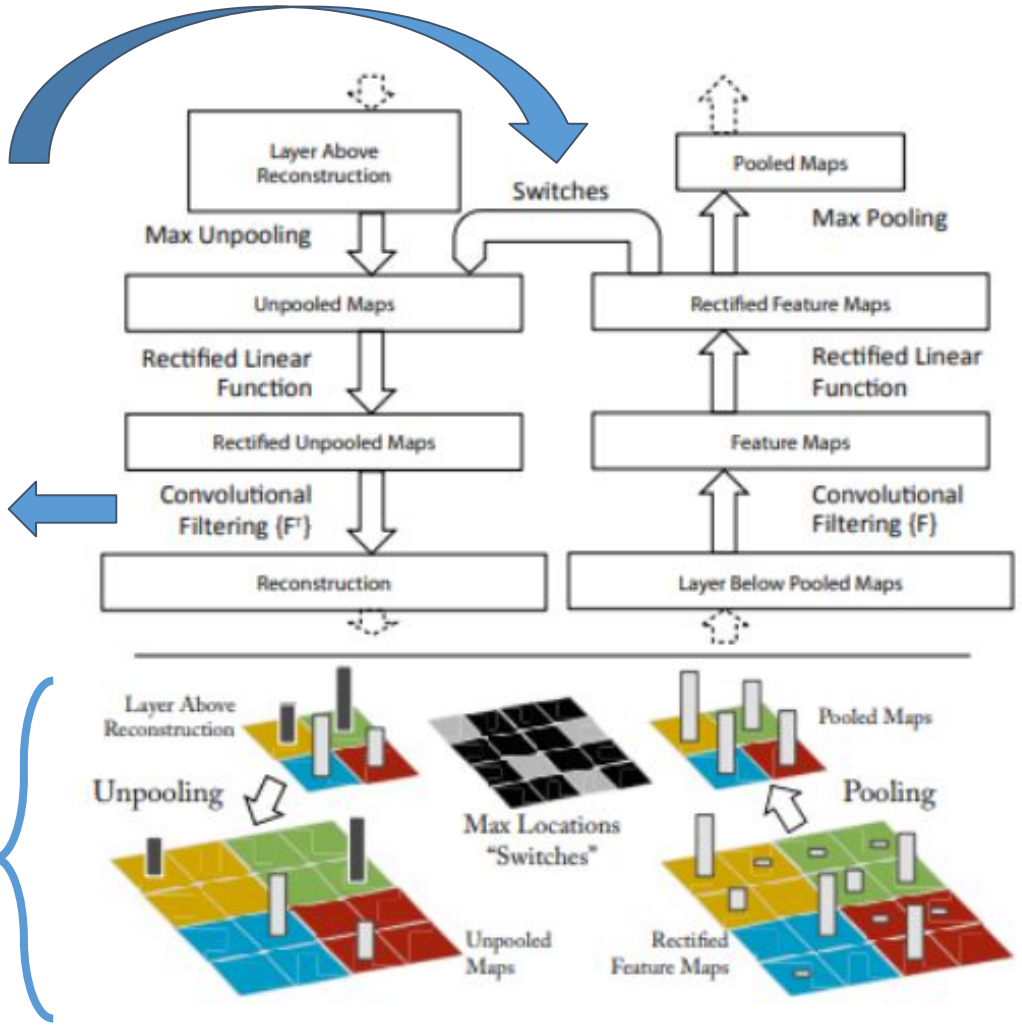


DeconvNet: unpooling ⇨ rectification ⇨ filtering

To keep edge information (avoid blurriness and color variation)

Transposed convolution

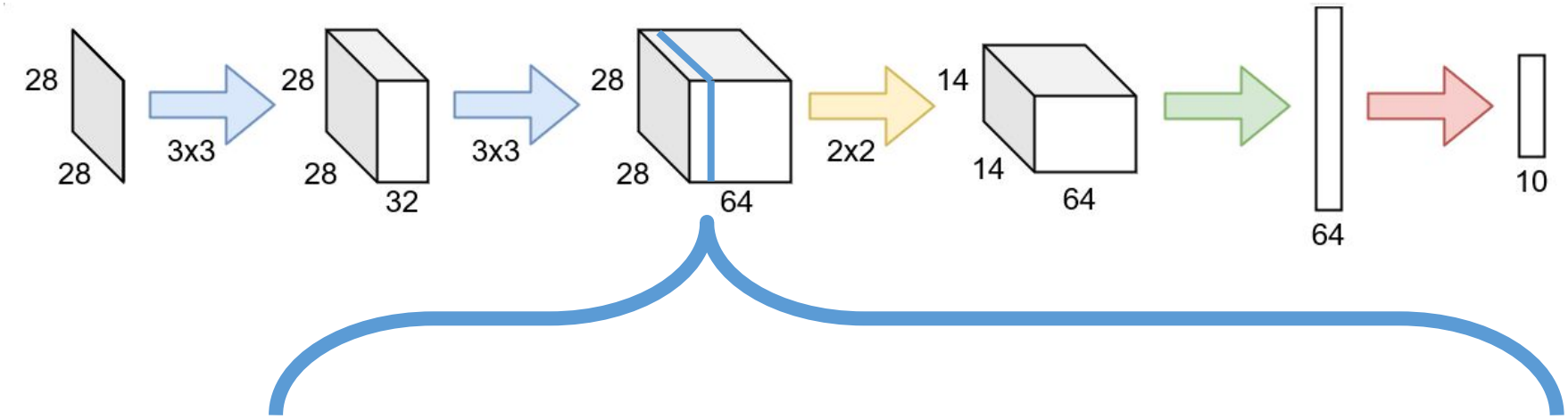




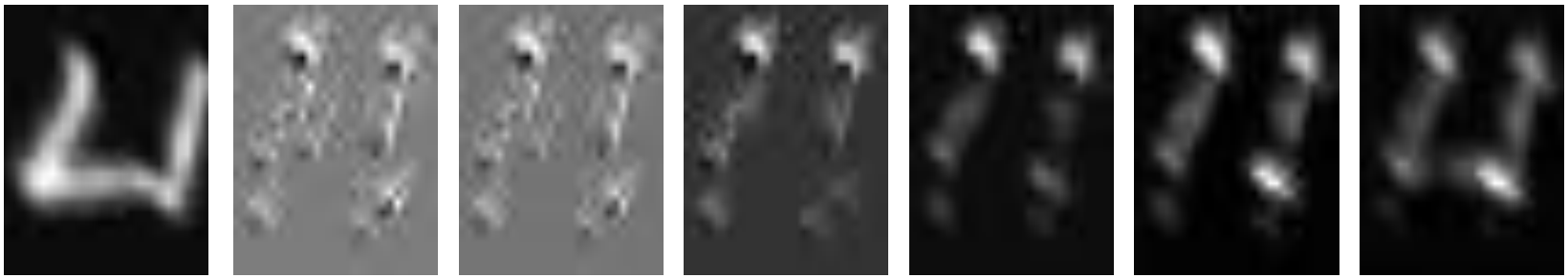
Reversible Max pooling, to get reconstruction clean

Longer bars higher activati-ons

Outcomes and results

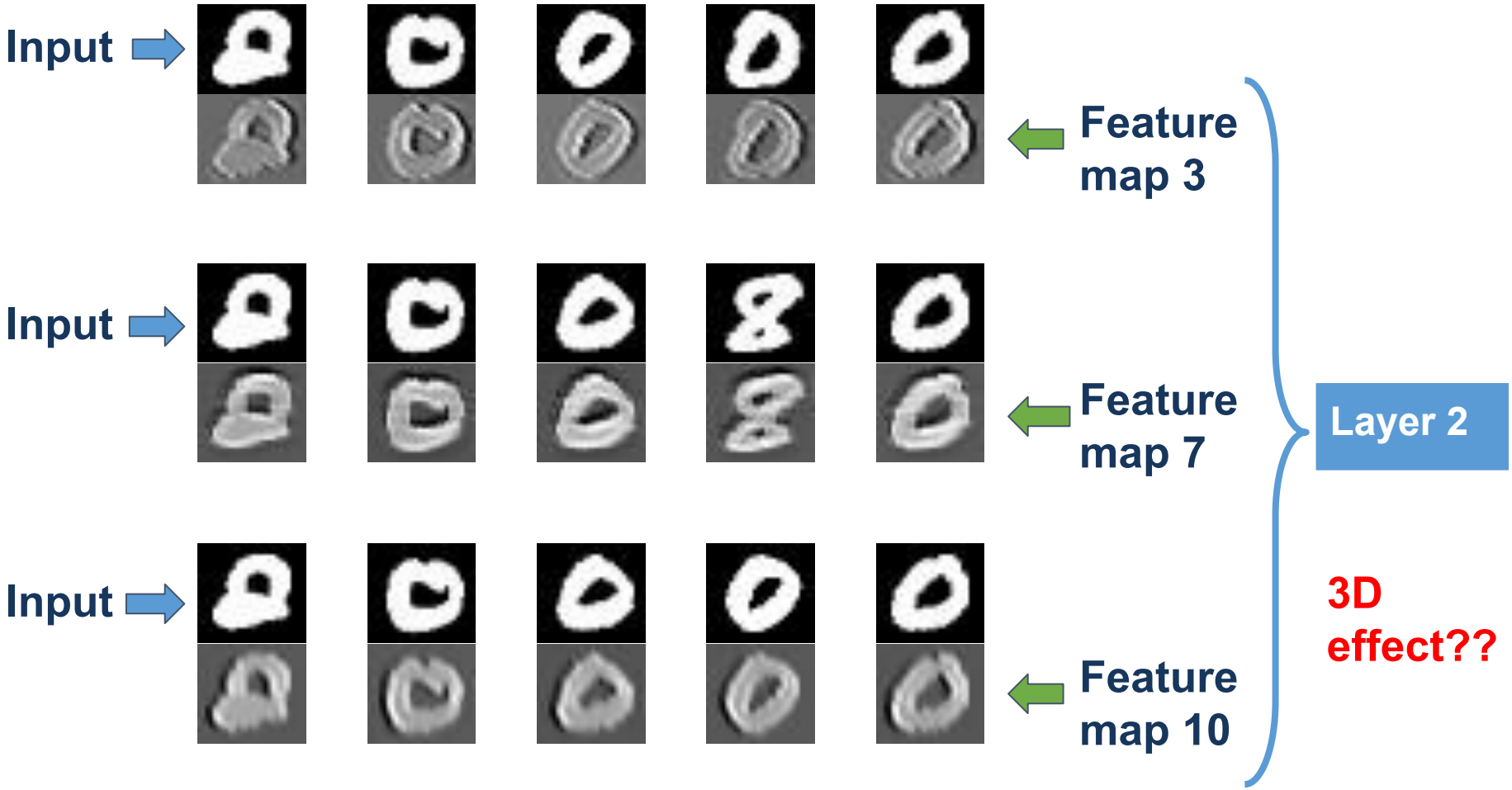


Visualisation during training (Layer 2, Feature Map 3):

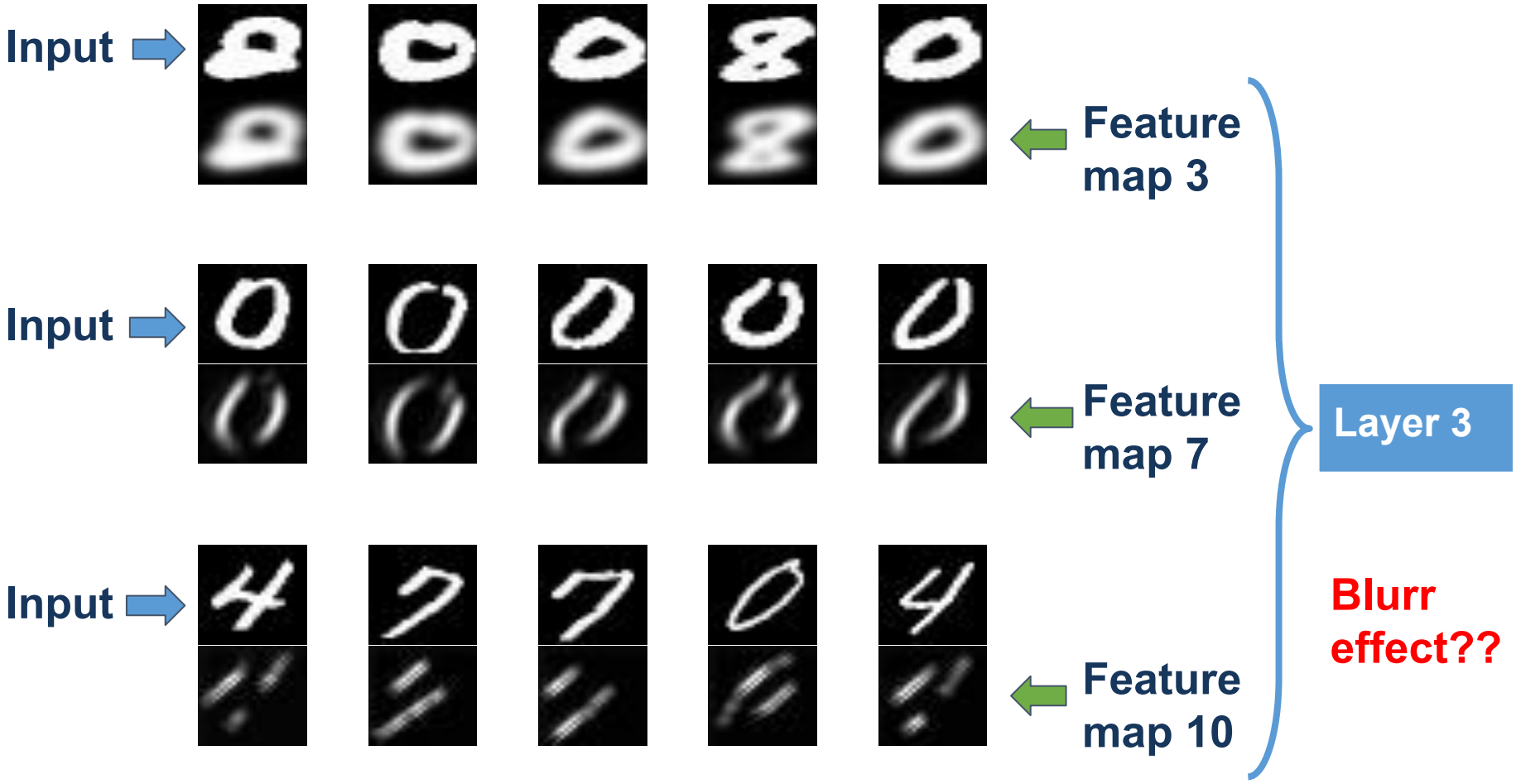


(input) (1 epoch) (2 epochs) (5 epochs) (10 epochs) (20 epochs) (50 epochs)

Visualization of top 5 activations

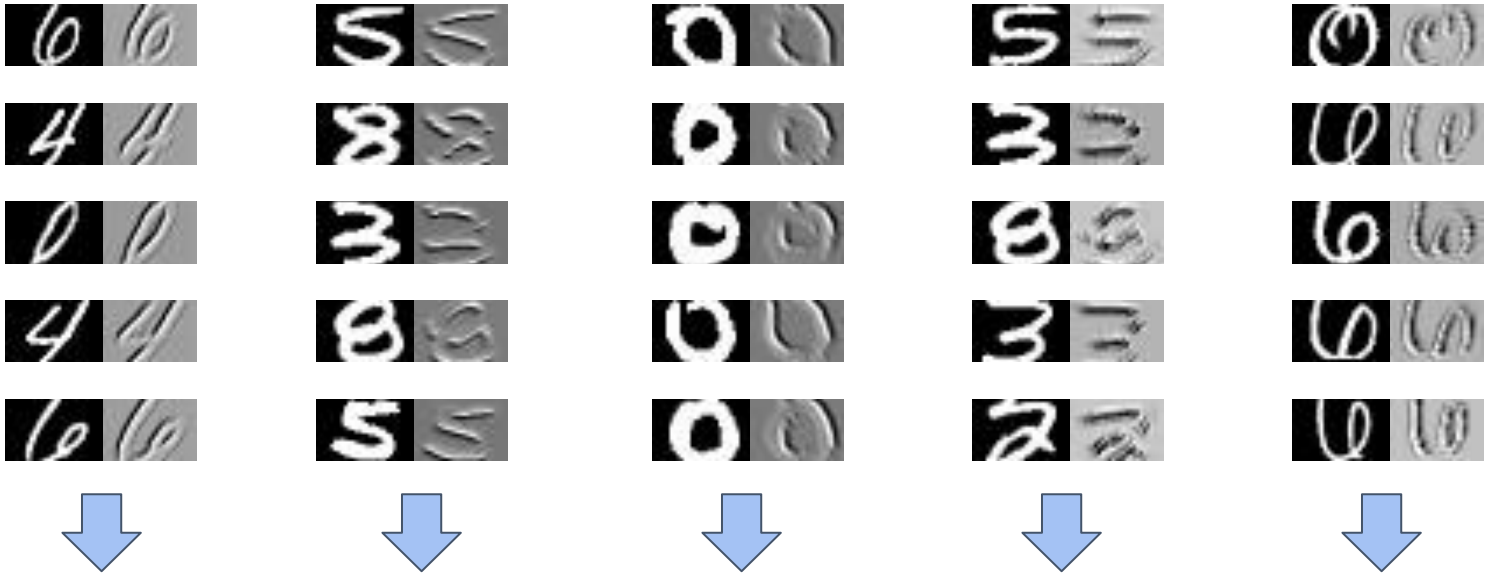


Visualization of top 5 activations



Top 5 activations on a deeper model

3D effect??



Layer 2
FM 3

Layer 2
FM 7

Layer 2
FM 10

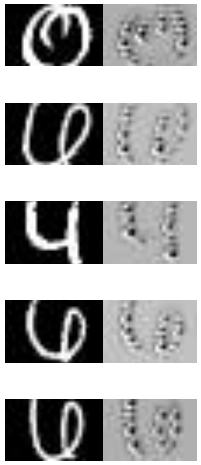
Layer 3
FM 3

Layer 3
FM 7

Top 5 activations on a deeper model

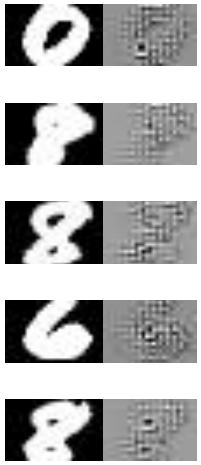
Clumping effect??

Layer 4
FM 3

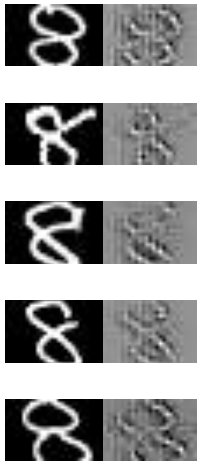


Layer 4
FM 7

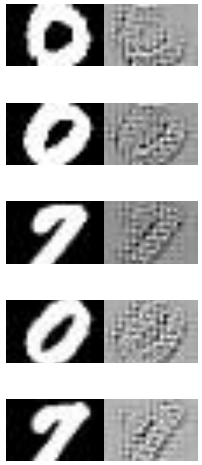
Crystal effect??



Layer 5
FM 3

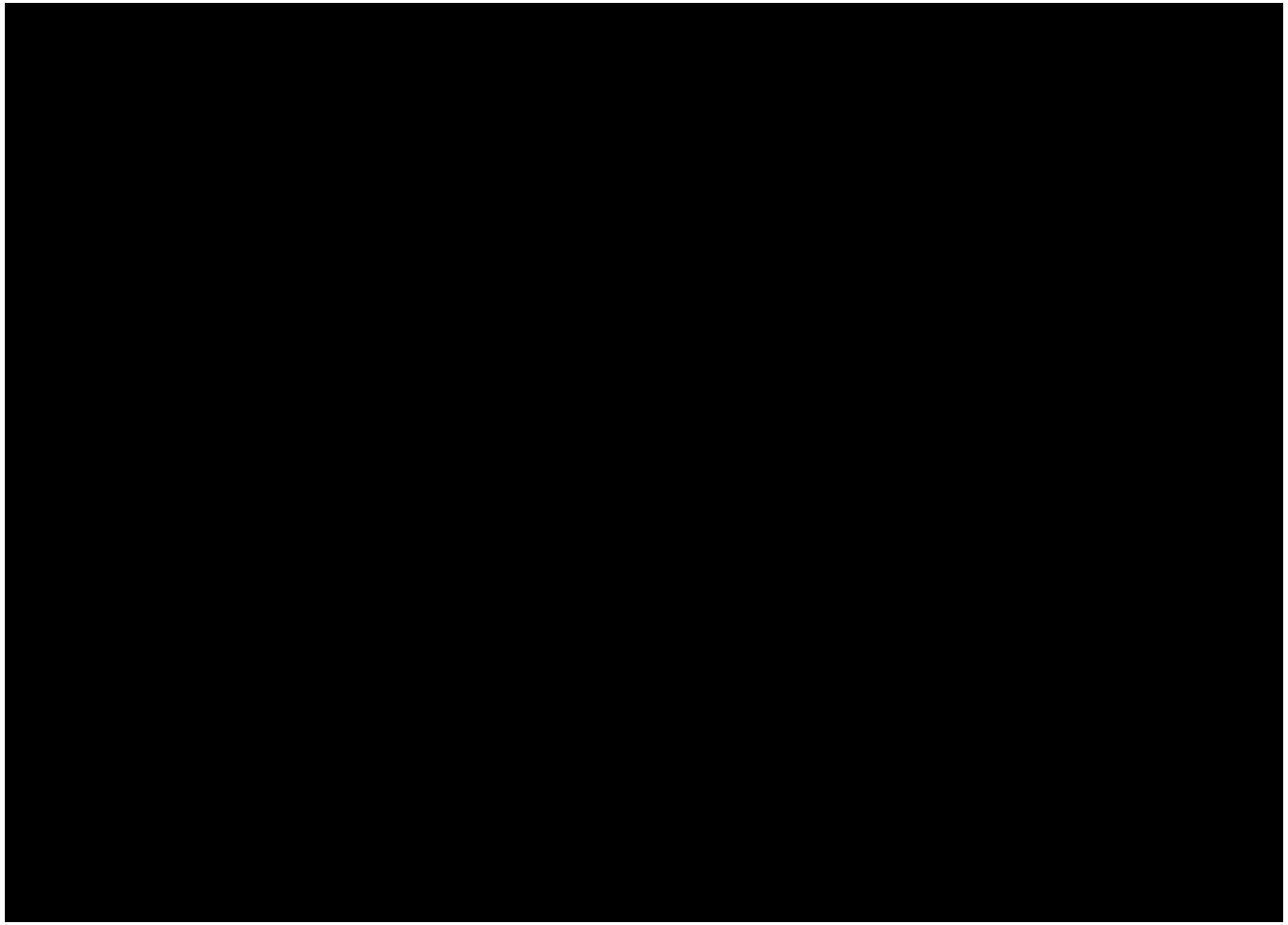


Layer 5
FM 7



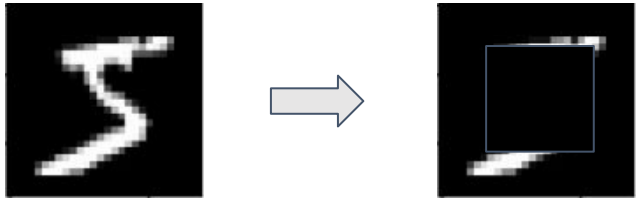
Layer 5
FM 10

Visualization of top 5 activations for RPS dataset



Sensitivity Analysis

- **Sensitivity analysis** of the classifier output by **occluding part of the input** image.
- **Drop** in **classification confidence** after occlusion.



```
60000/60000 [==============================] - 170s 3ms/sample - loss: 0.0051 - accuracy: 0.9994 - val_loss: 0.1577 - val_accuracy: 0.9924
current directory:  /home/hkwak/Documents/Workspace/cnn-make-training-visible/curtiz_MNIST_deconv_model3_2020-01-26_16:44:33
Saved trained model at /home/hkwak/Documents/Workspace/cnn-make-training-visible/curtiz_MNIST_deconv_model3_2020-01-26_16:44:33/saved_models
model_func1 prediction: normal, blacked
model_func1.predict(normal_img):  [[0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]]
model_func1.predict(img_blacekd):  [[1.7228854e-15 1.5951092e-14 6.5399600e-12 1.5834305e-28 3.8626182e-03
  0.0000000e+00 0.0000000e+00 9.9613732e-01 1.6871501e-26 4.7516211e-28]]
10000/10000 [==============================] - 3s 250us/sample - loss: 0.1577 - accuracy: 0.9924
```

Reference:

Matthew D Zeiler, Rob Fergus, Visualizing and Understanding Convolutional Networks (2013)

Our Project on GitHub:

Questions?