

- GitHub Copilot / ChatGPT to help writing code
- MS Copilot to generate poster illustration

References:

Lenail, Alex. *NN-SVG: LeNet*. <http://alexlenail.me/NN-SVG/LeNet.html>,
Paszke, Adam, et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019,
<https://pytorch.org/>.

HotdogNet

Finding the best hotdog classifier

Group 21: Zeljko Antunovic (s233025), Alex Belai (s233423), Lukas Samuel Czekalla (s233561), Filip Penzar (s232452), Nándor Takács (s232458)

Data

Origin of the data:

- Subset of ImageNet-Dataset
- *HOTDOG* = chilidog & hotdog classes
- *NO-HOTDOG* = pets, furniture, people, food & frankfurter classes

Initial data transformation:

- Resizing to 256×256 px
- Normalization w.r.t. ImageNet mean & std

Data Composition:

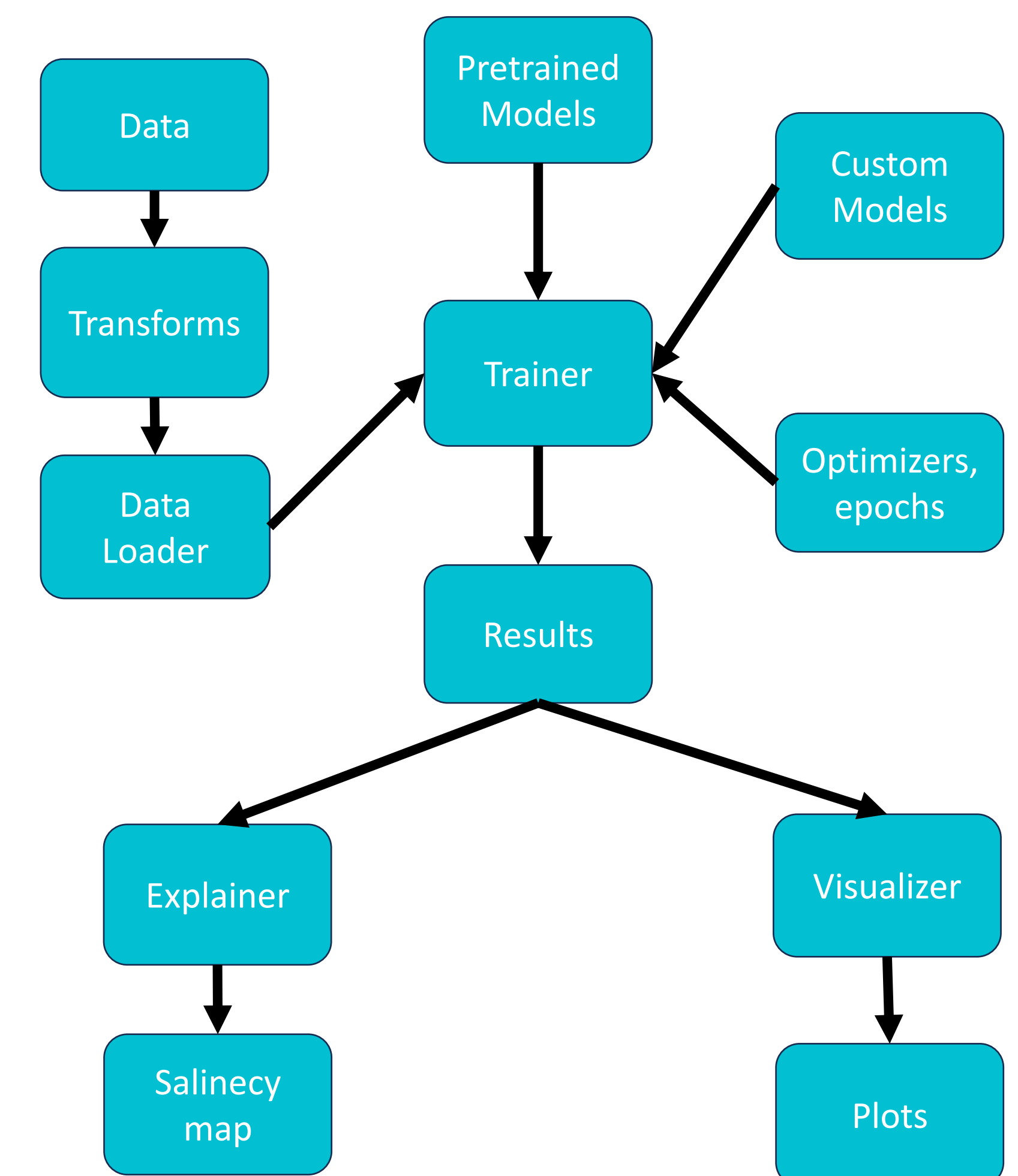
Training-data		Test-data	
hotdog	no-hotdog	hotdog	no-hotdog
1075	972	895	967

- Small amount of images = risk of overfitting



Examples of training images. The lower images are normalized.

Implementation

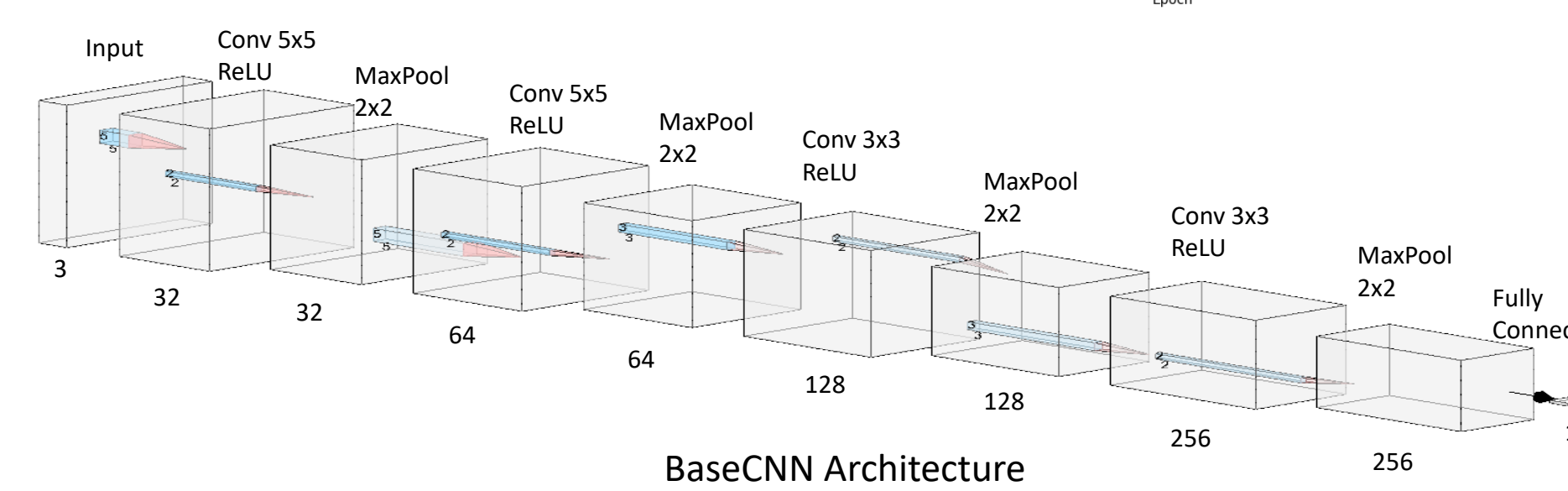


Experiments

Baseline setup

- One output neuron with sigmoid
- Loss Function = Binary Cross Entropy
- Optimizer = Adam with $l_r = 10^{-4}$
- No X-Val. Or 3rd validation set
- Training for 25 epochs
- Comparison of different architectures

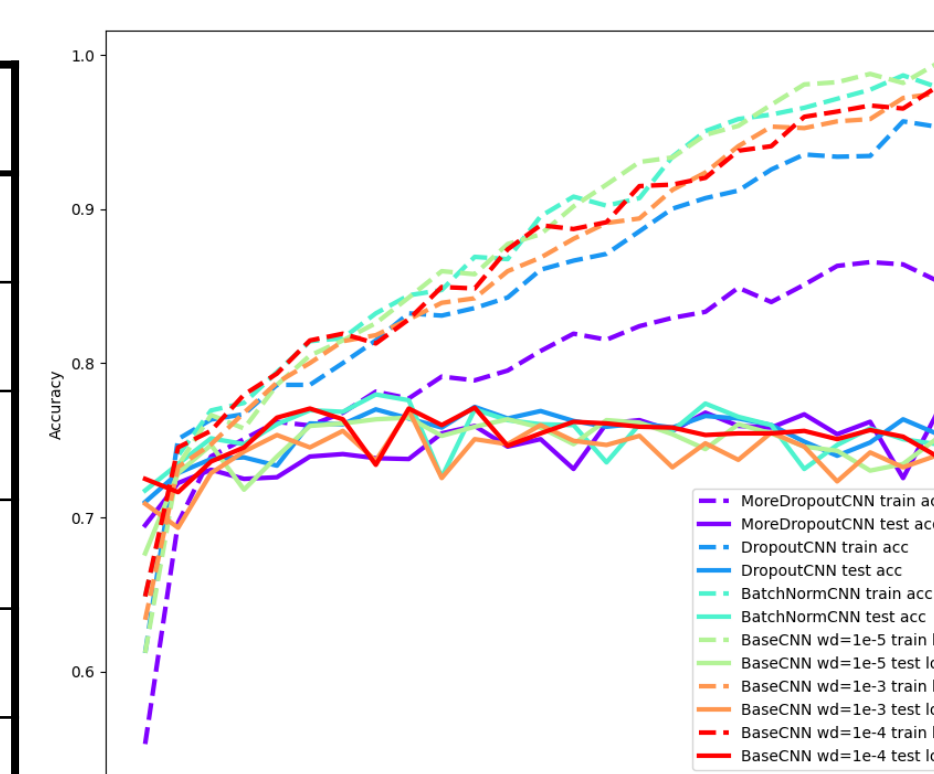
Architecture	Test-Acc.
BaseCNN	75.5%
VGG11	79.8%
ResNet18	78.0%
DenseNet121	80.1%



Regularization

- Modification of BaseCNN architecture

Method	Test-Acc.
Dropout $p = 0.05$	75.4%
Dropout $p = 0.2$	76.6%
Batchnorm	74.8%
WD $w = 10^{-3}$	74.0%
WD $w = 10^{-4}$	74.0%
WD $w = 10^{-5}$	75.1%

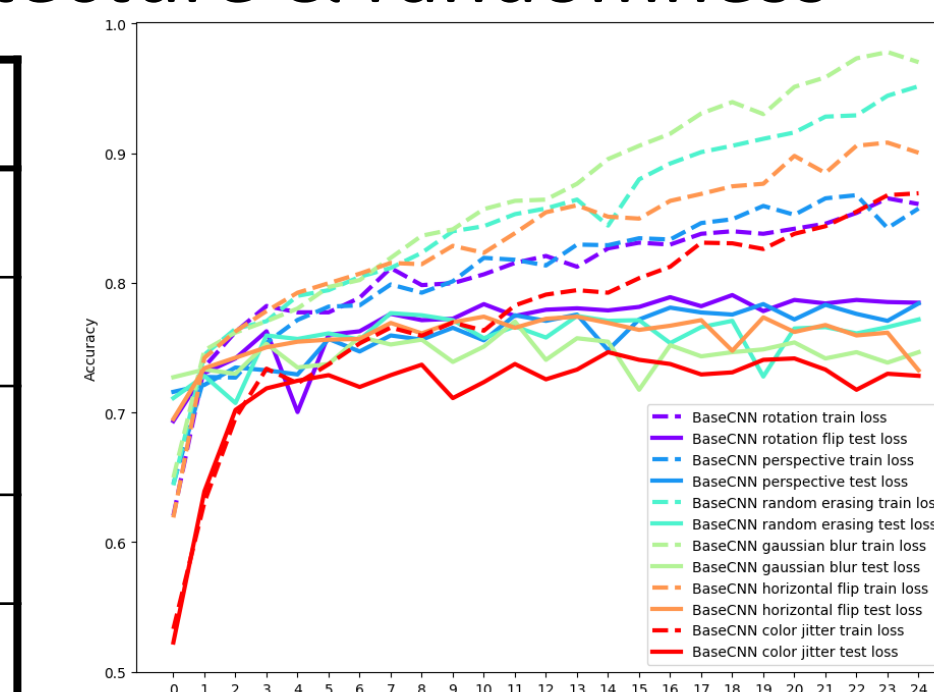


- Dropout is promising

Data Augmentation

- Using BaseCNN architecture & randomness

Method	Test-Acc.
Horizontal Flip	73.2%
Rotation	78.5%
Perspective	78.4%
Color Jitter	72.8%
Gaussian Blur	74.7%
Random Erasing	77.2%

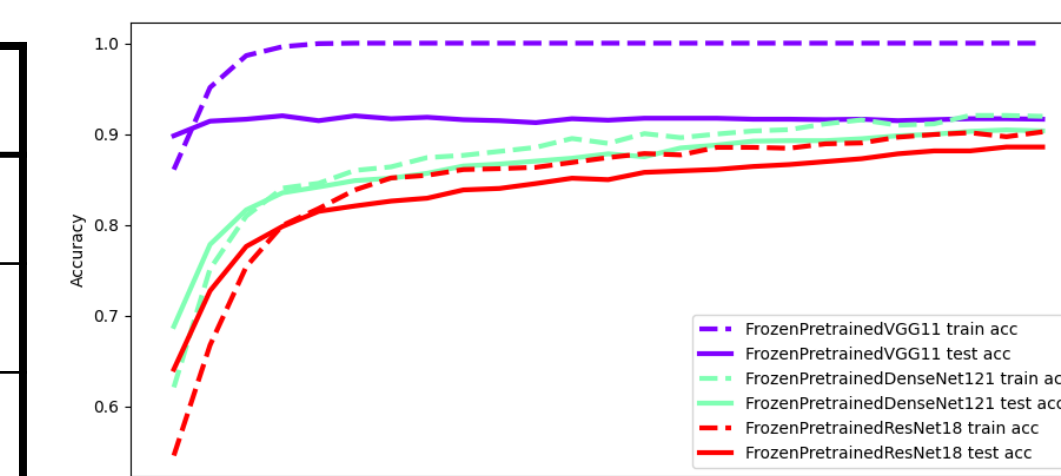


- Various methods suitable

Transfer Learning

- Pretrained classifiers on ImageNet-data
- Convolutional layers are frozen
- Retraining of fully-connected layers

Model	Test-Acc.
VGG11	91.6%
ResNet18	88.6%
DenseNet121	90.3%

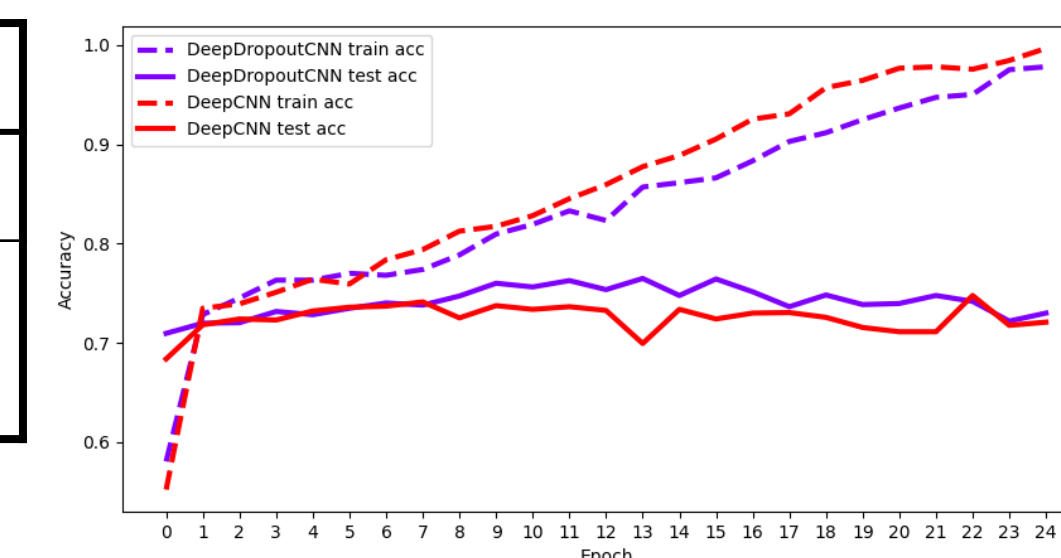


- ResNet & DenseNet don't show overfitting

Increase depth of model

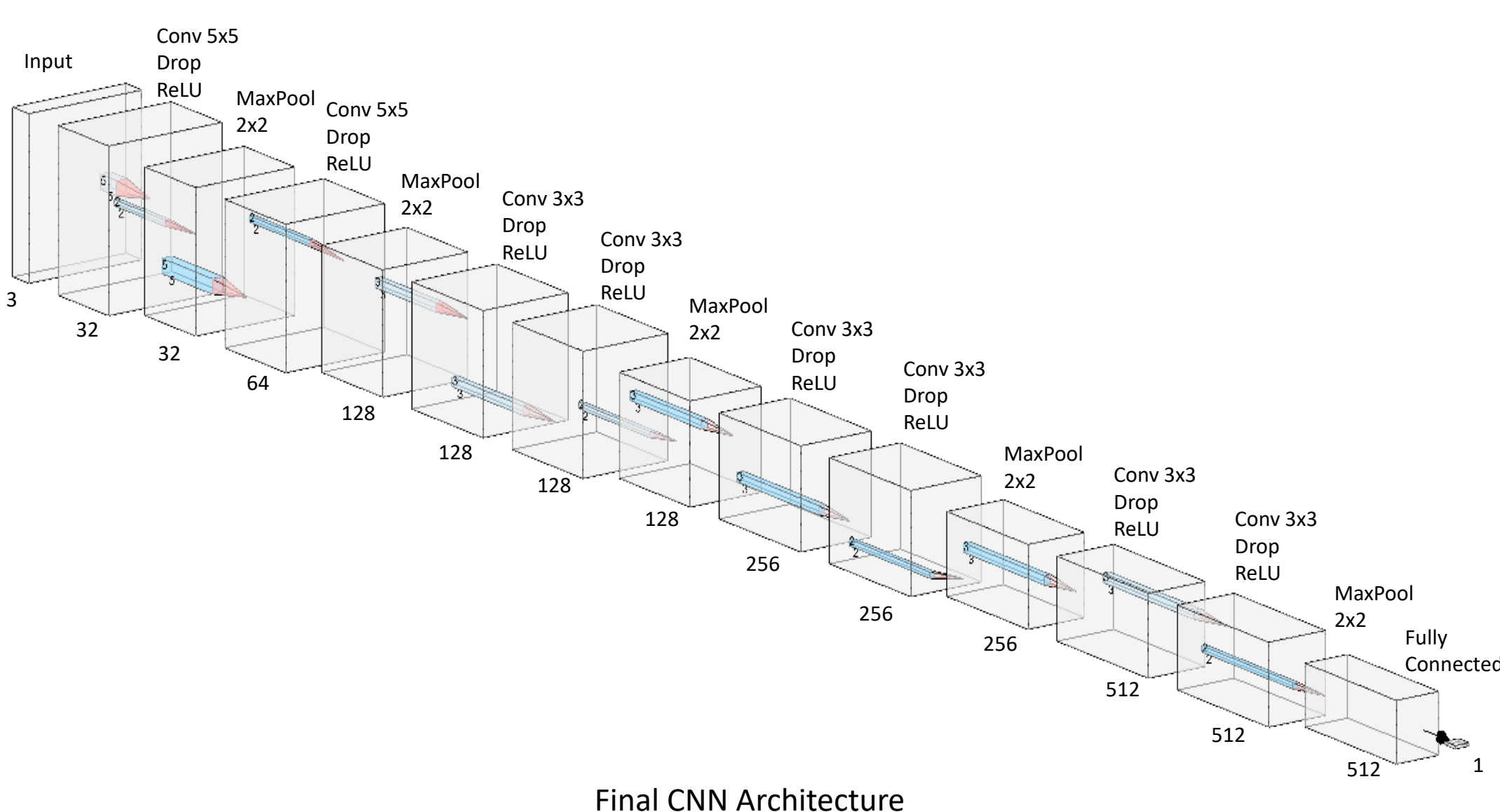
- BaseCNN with $2 \times$ amount of conv. layers

Method	Test-Acc.
$2 \times$ conv. layers	72.1%
$2 \times$ conv. layers + dropout	73.0%



- Stronger overfitting

Final Architecture



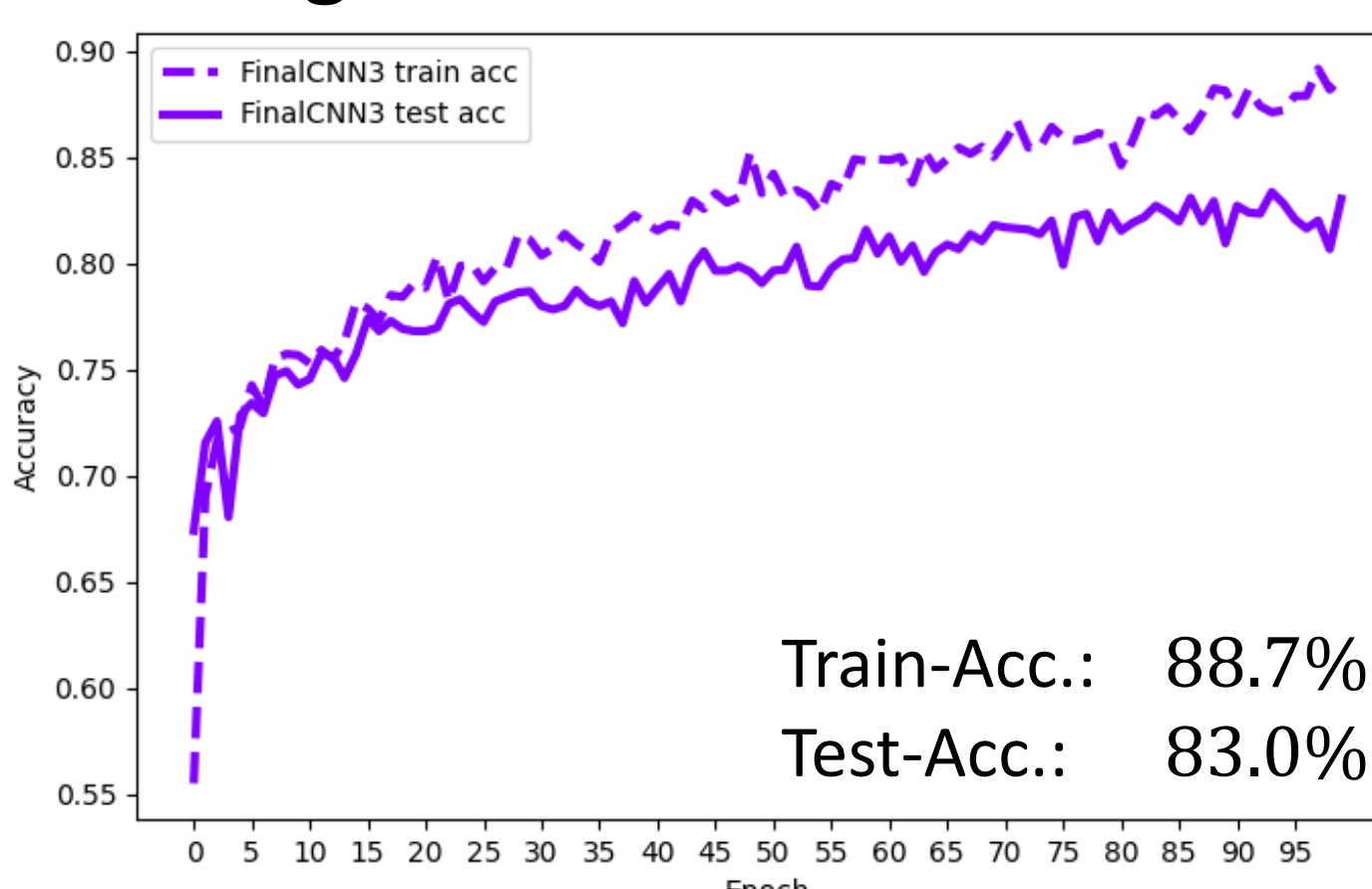
Final CNN Architecture

Architecture Considerations

- Convolutional layers increasing channels $\times 2$
- Kernel size in conv. layers: 5×5 and 3×3
- Max-Pooling (2×2) reduces spatial dim. by $\times 2$
- Dropout (0.1) & weight-decay (10^{-5}) against overfitting
- One fully-connected layer

Interpretation & Result

Training results of final arch.

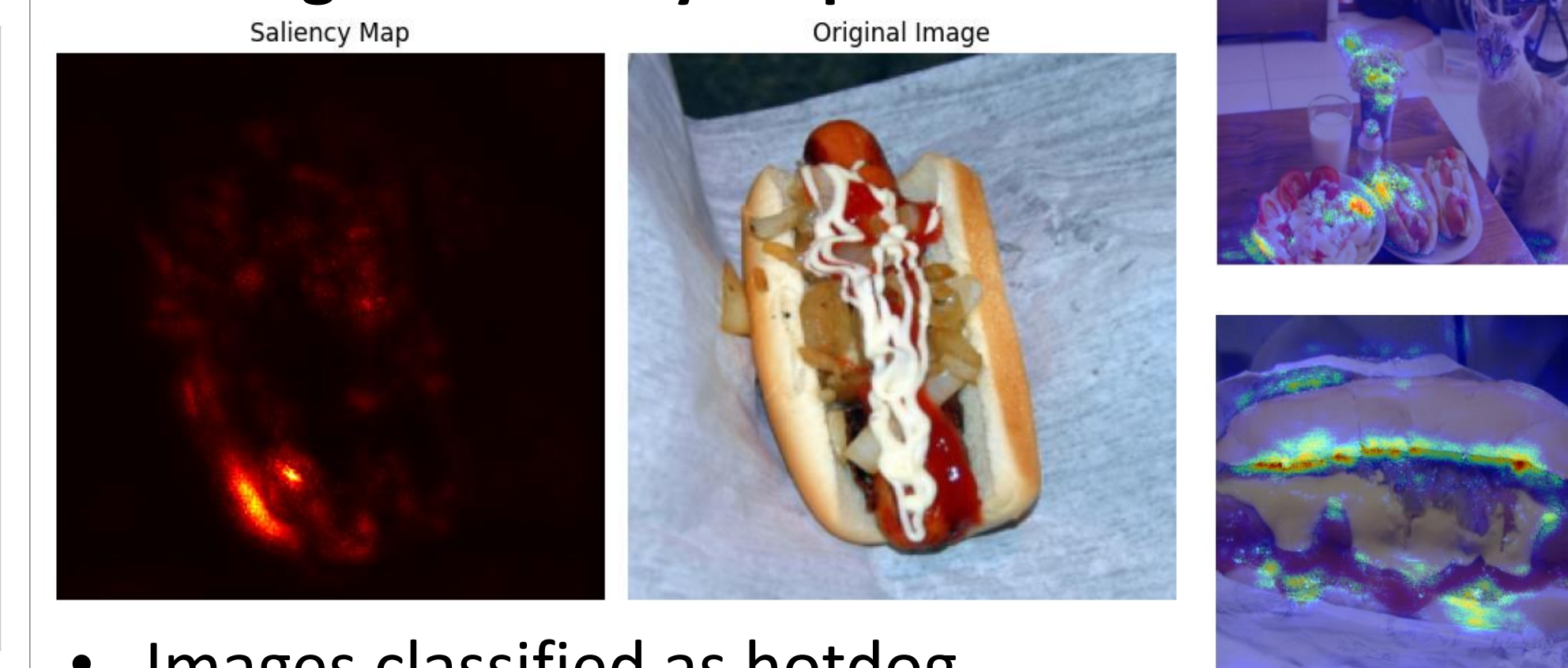


Train-Acc.: 88.7%
Test-Acc.: 83.0%

Confusion Matrix

True Label	Predicted Label	
	Hotdog	No Hotdog
Hotdog	768	127
No Hotdog	189	778

Smoothgrad Saliency Map



- Images classified as hotdog

Missclassified images

Top 3 misclassified images with highest loss

