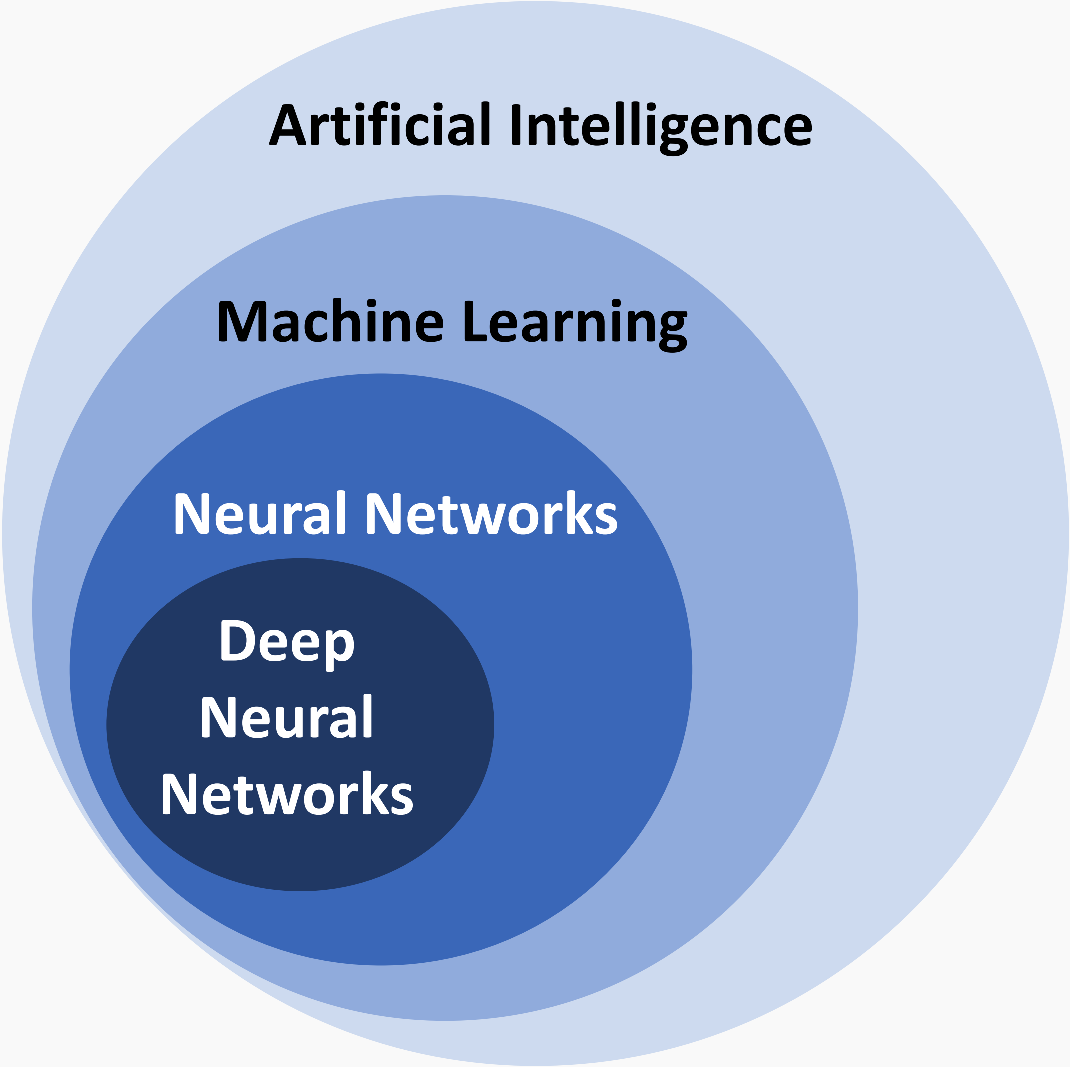
Convolutional Neural Networks

Tamás Gábor

Seminar KF3

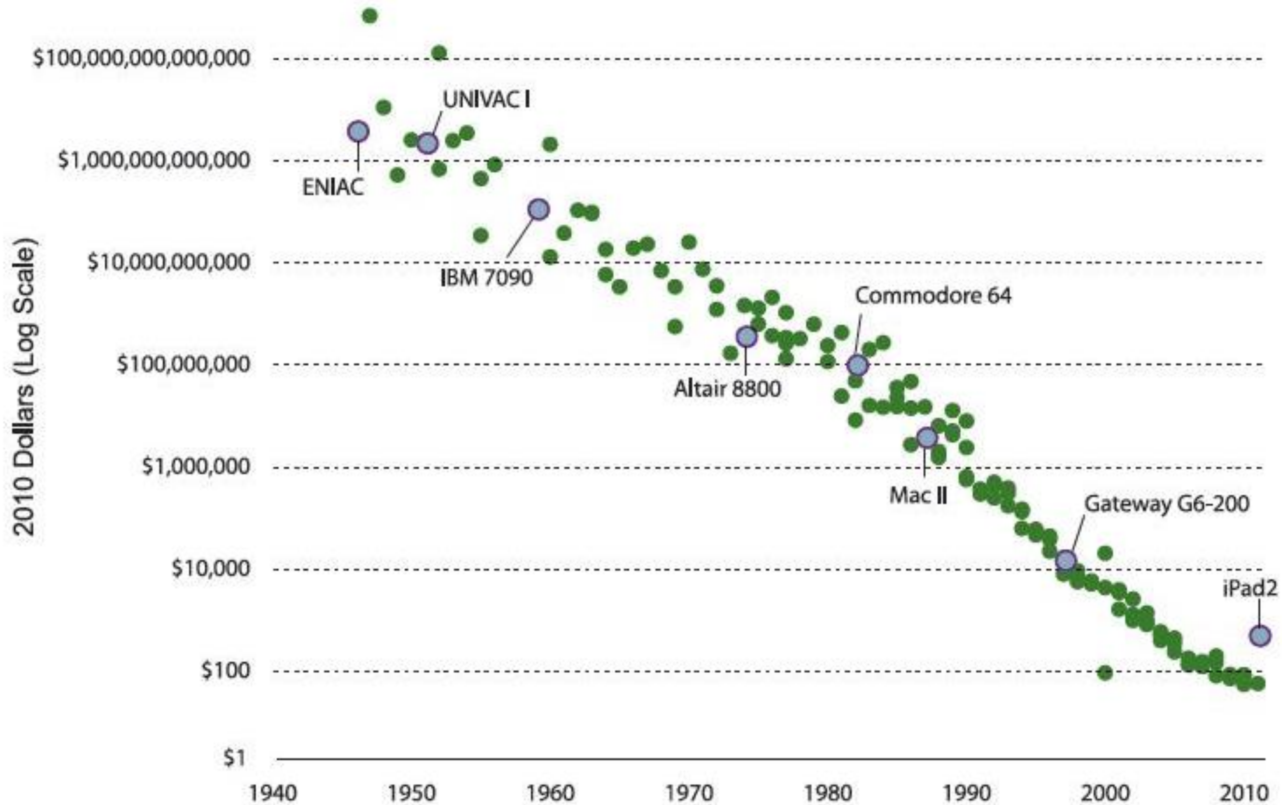
Budapest University of Technology and Economics
27.09.2019

Overview

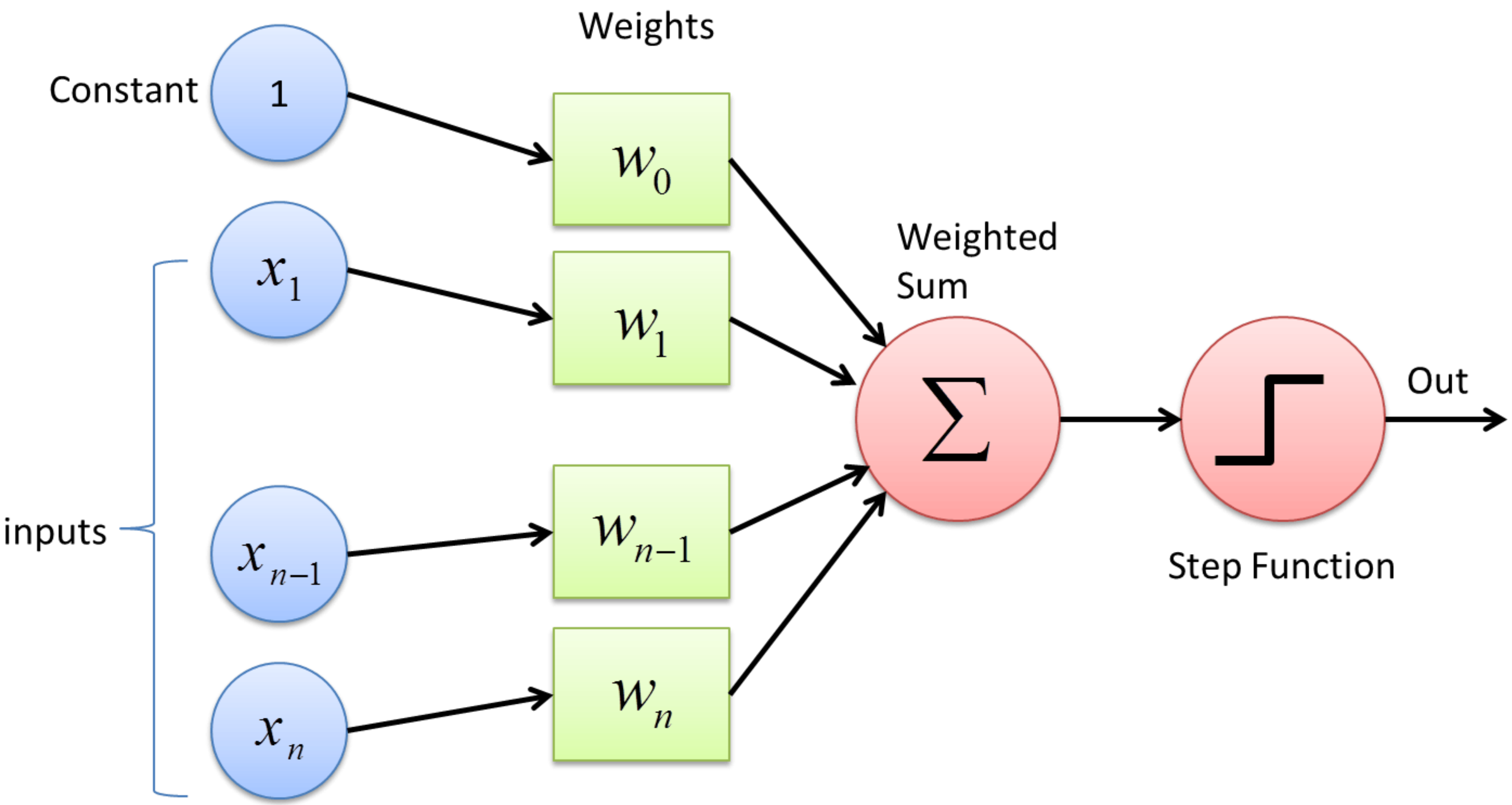

Artificial Intelligence
Machine Learning
Neural Networks
Deep Neural Networks

The rise of neural networks

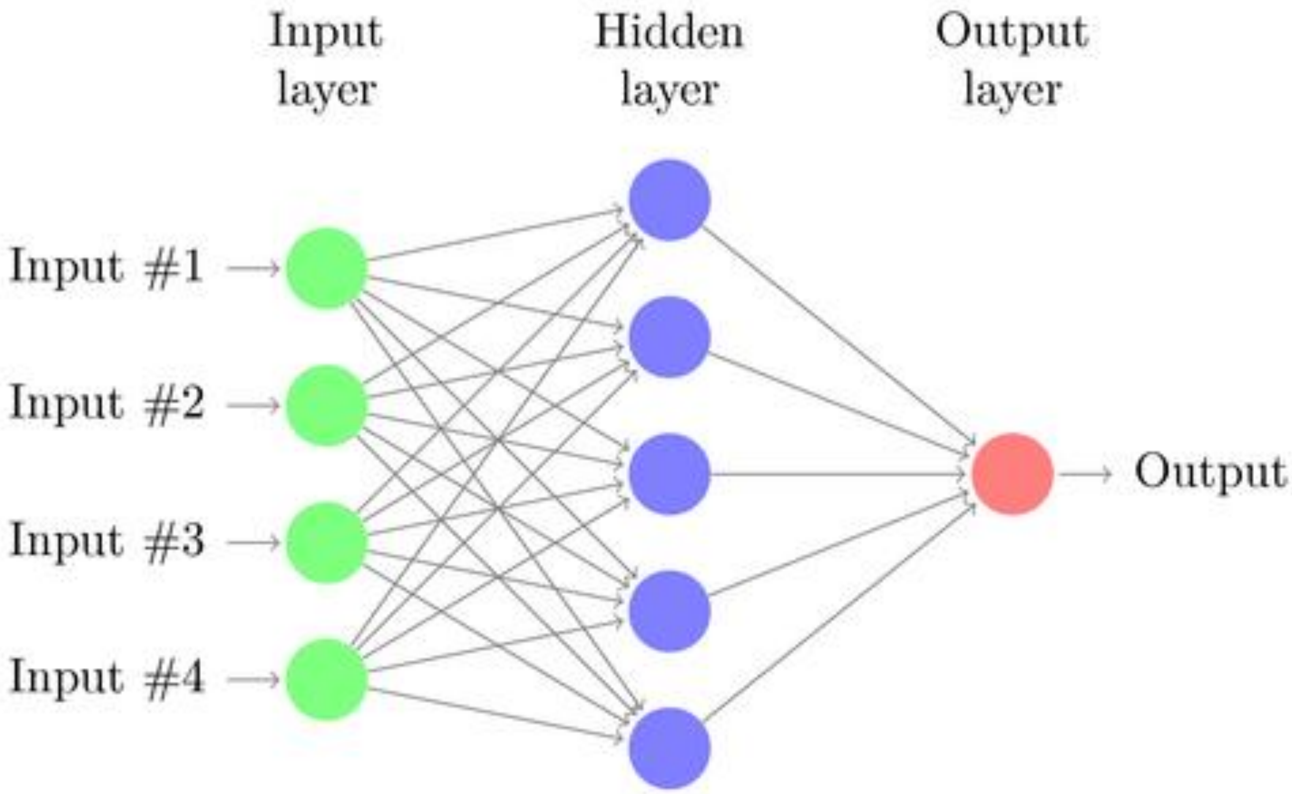
- Growing availability of cheap cloud computing and GPUs
- Large amounts of data, new open-source tools



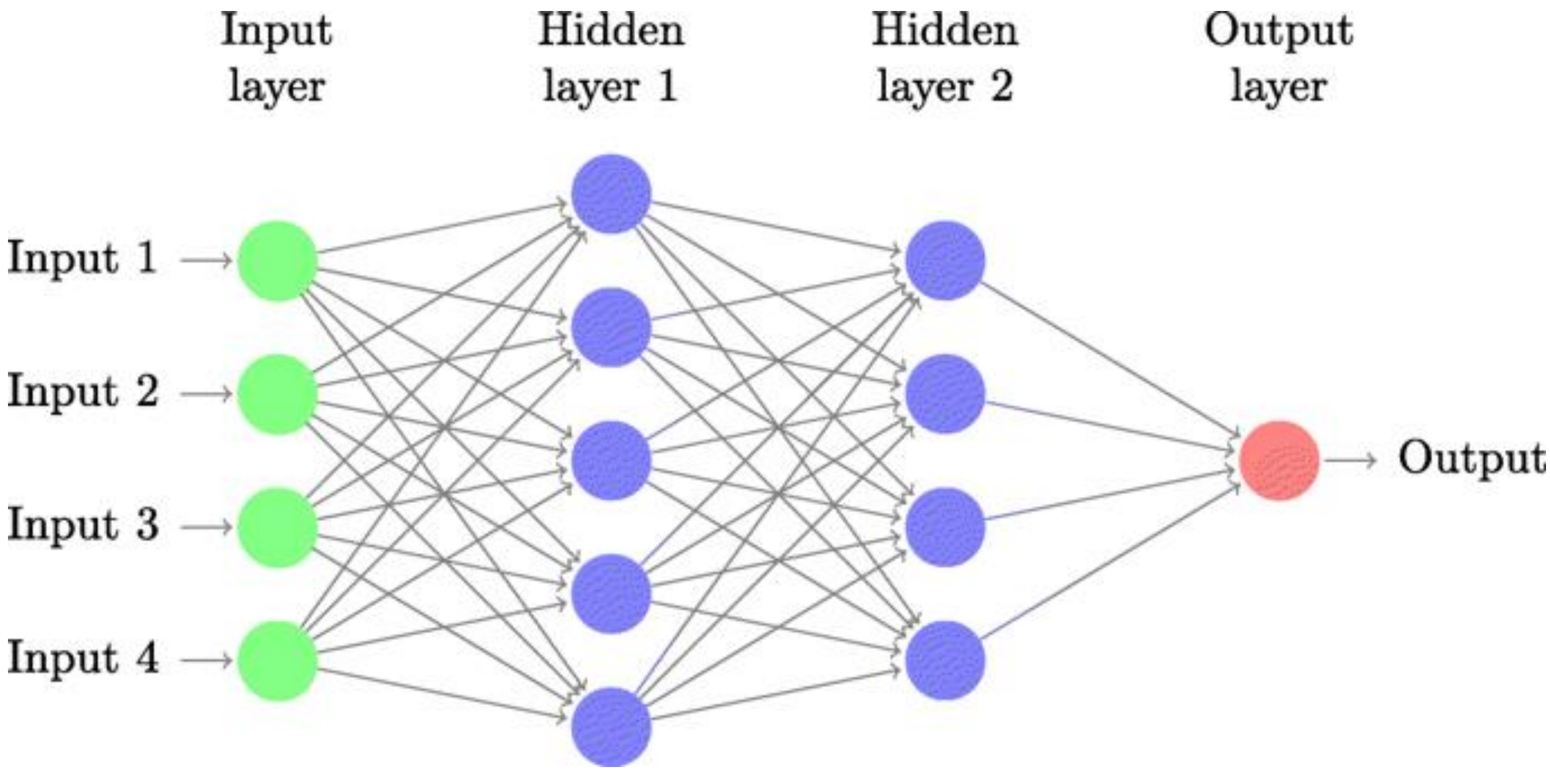
Perceptron


Weights
Constant
1
w_0
x_1
w_1
Weighted
Sum
Σ
Out
inputs
x_{n-1}
w_{n-1}
Step Function
x_n
w_n

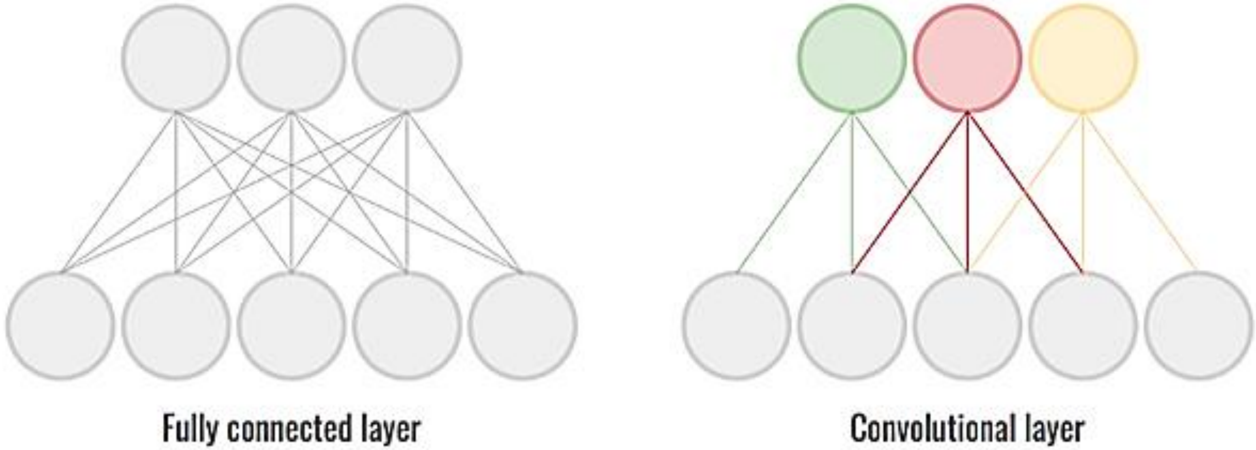
Artificial neural network – 1 hidden layer

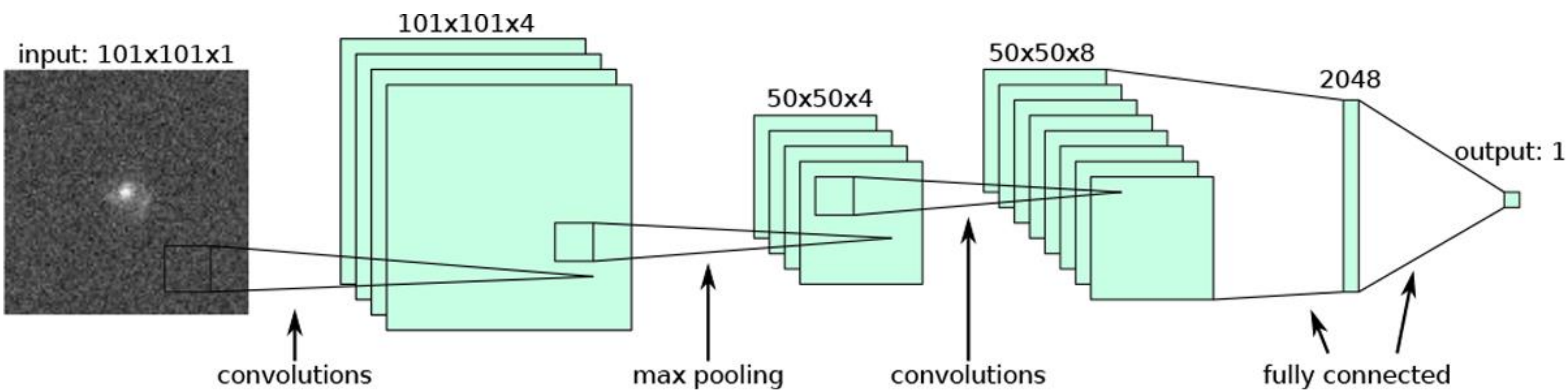


Artificial neural network – 2 hidden layers

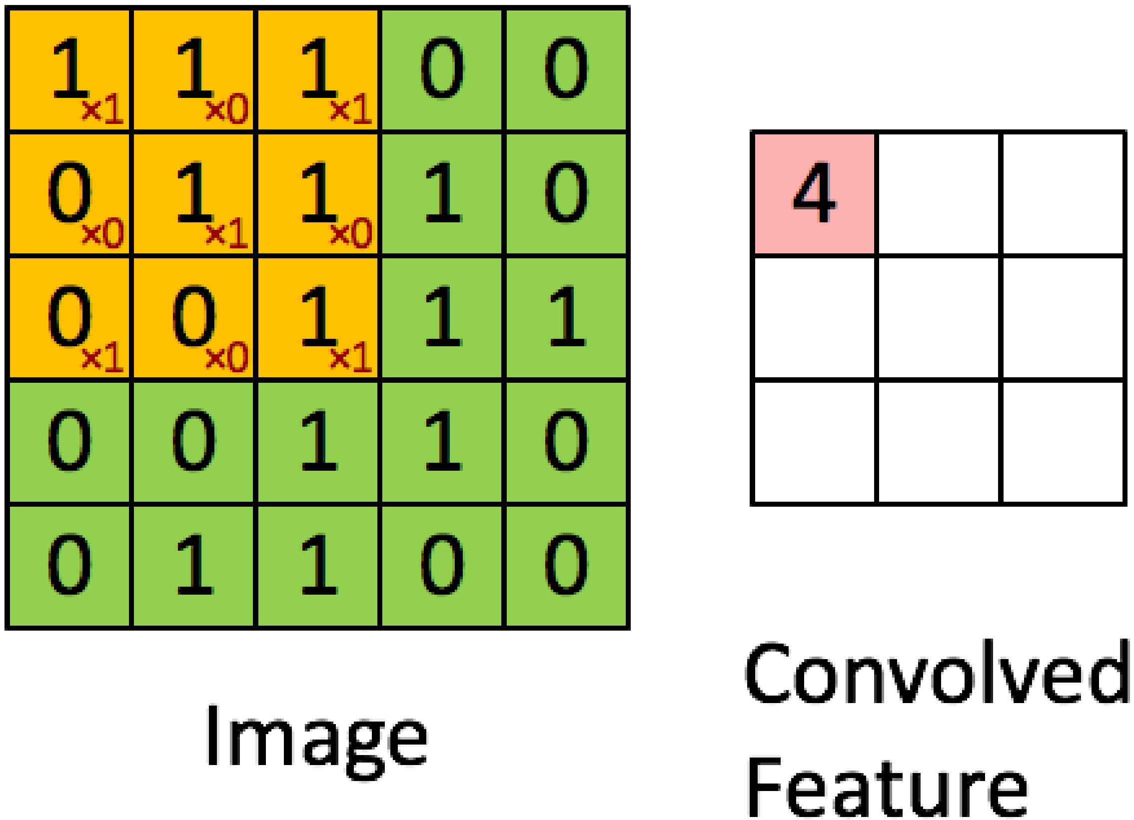


Deep Convolutional Neural Networks

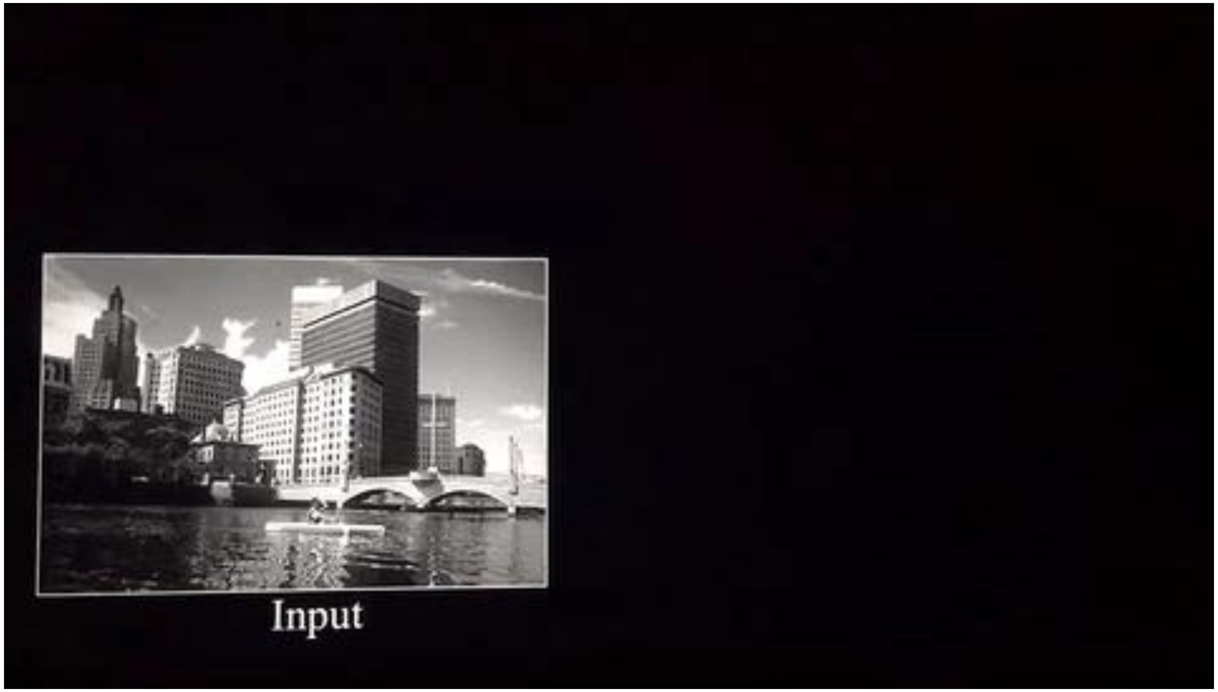




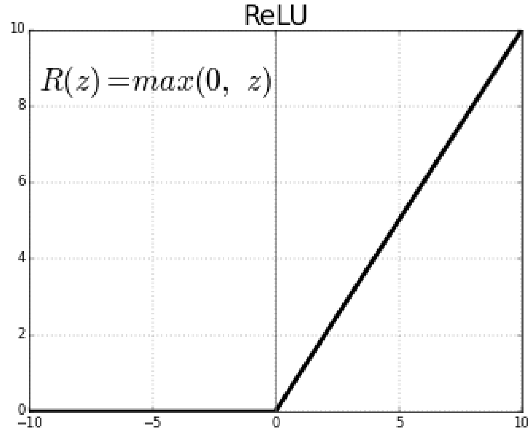
Convolutional layers

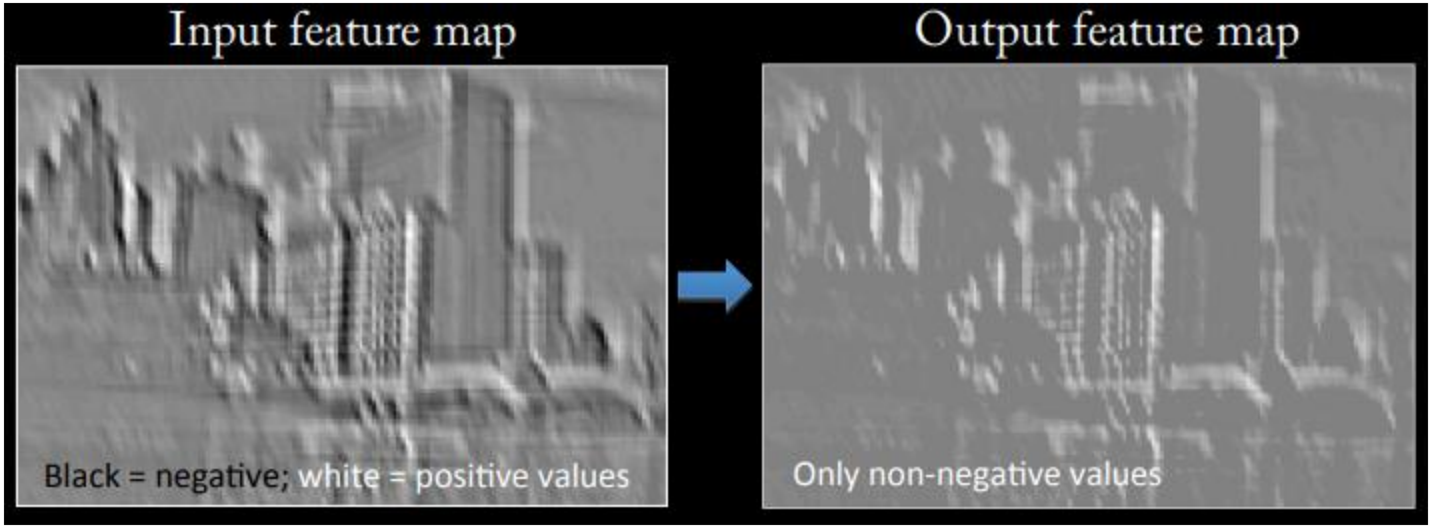


Convolutional layers

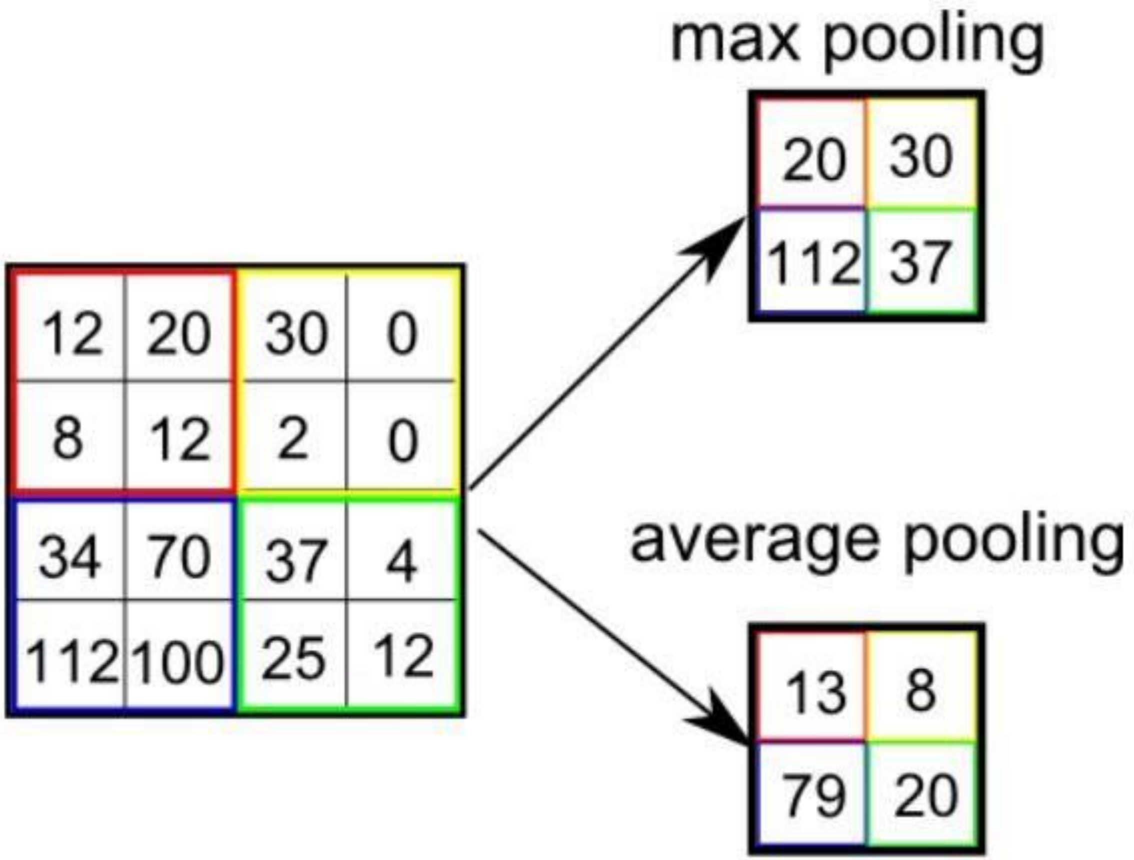


Activation - ReLU



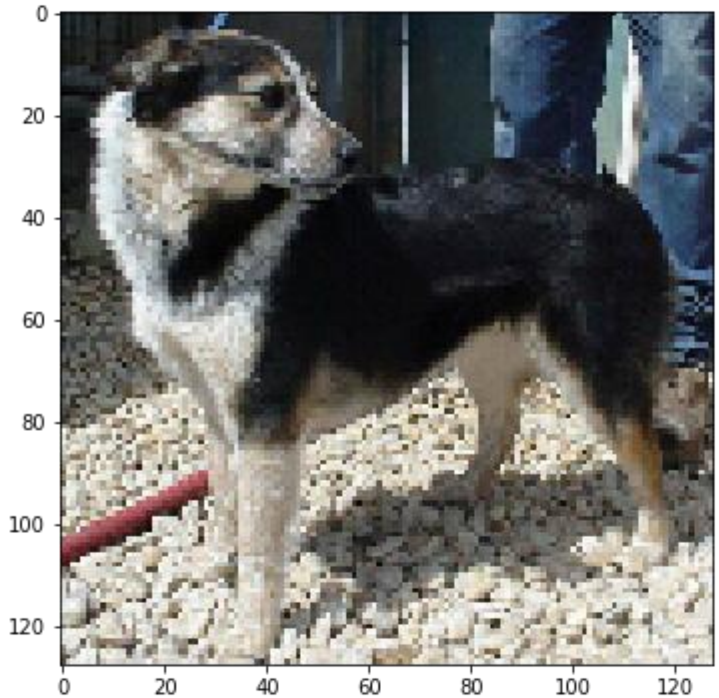


Pooling


max pooling
average pooling
12
20
30
0
8
12
2
0
34
70
37
4
112
100
25
12
20
30
112
37
13
8
79
20

Cats vs dogs with a CNN

- Supervised learning
- 8000 pictures for training



Train-validation-test split

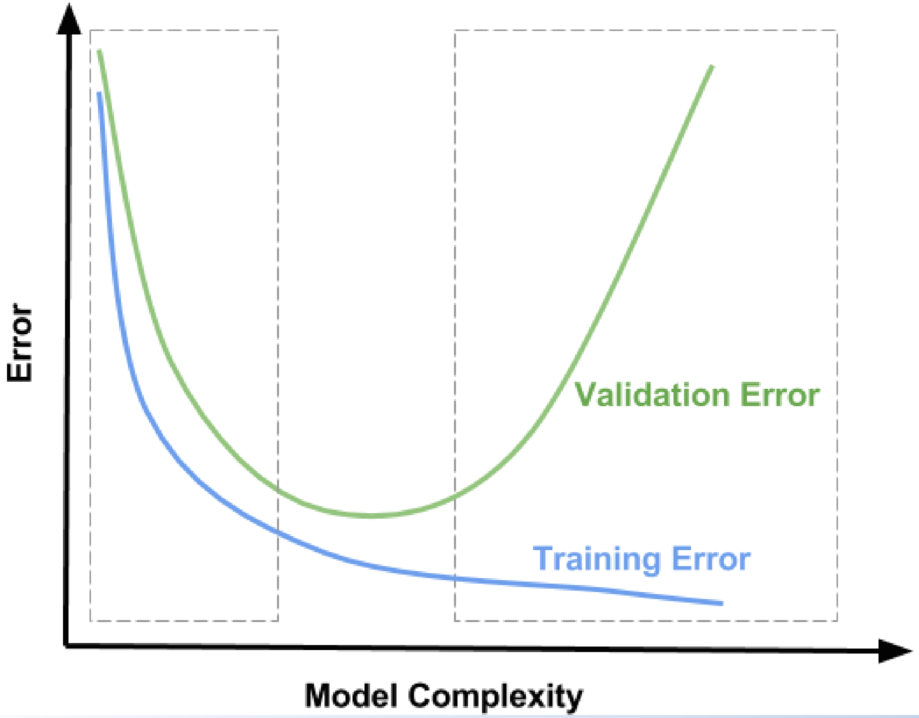
- Training dataset *(6400 pictures)*
  The sample of data used to fit the model.

- Validation dataset *(1600 pictures)*
  Provides an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters.

- Test dataset *(2000 pictures, unknown labels)*
  Provides an unbiased evaluation of a final model fit on the training dataset.

Difficulties

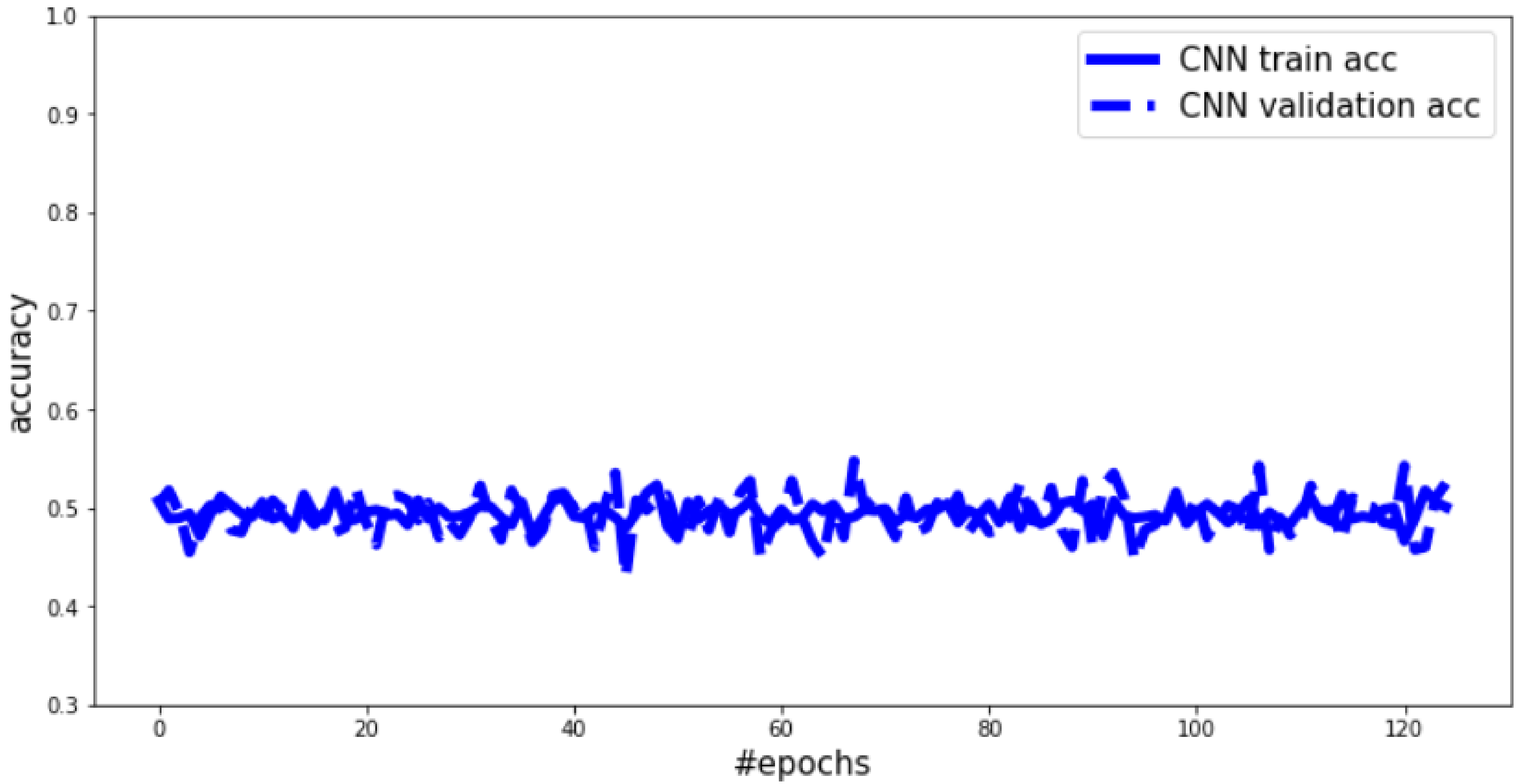
- Large hyperparameter search space
- Small training dataset -> data augmentation
- Overfitting



Simple CNN for image classification

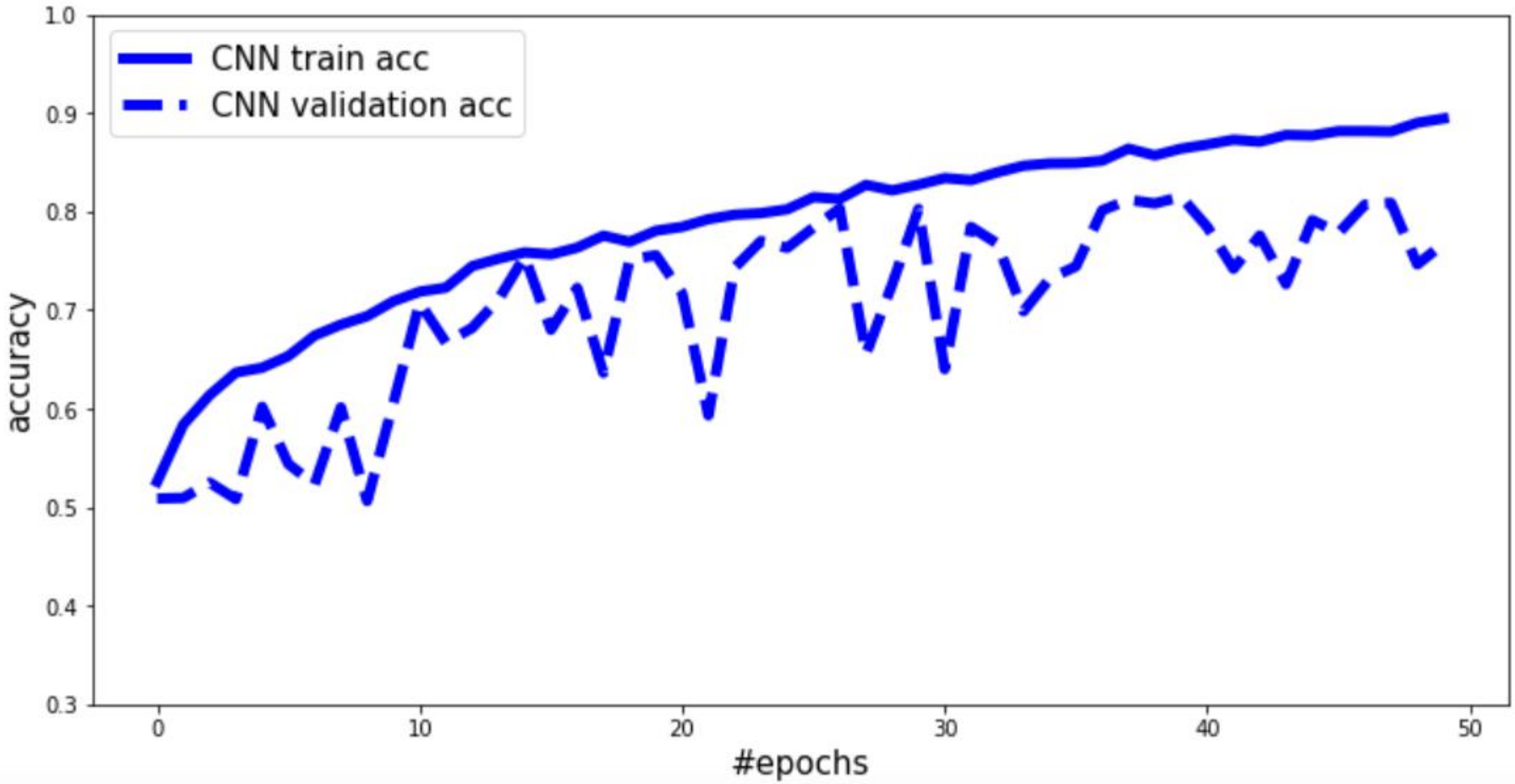
```
cnn_model = Sequential()

cnn_model.add(Conv2D(32, (3, 3), use_bias=False, input_shape=input_shape))
cnn_model.add(Activation("relu"))
cnn_model.add(MaxPooling2D((2, 2)))

cnn_model.add(Flatten())
cnn_model.add(Dense(100, use_bias=False))
cnn_model.add(Activation("relu"))
cnn_model.add(Dense(1, activation='sigmoid'))
```

Accuracy for different network architectures

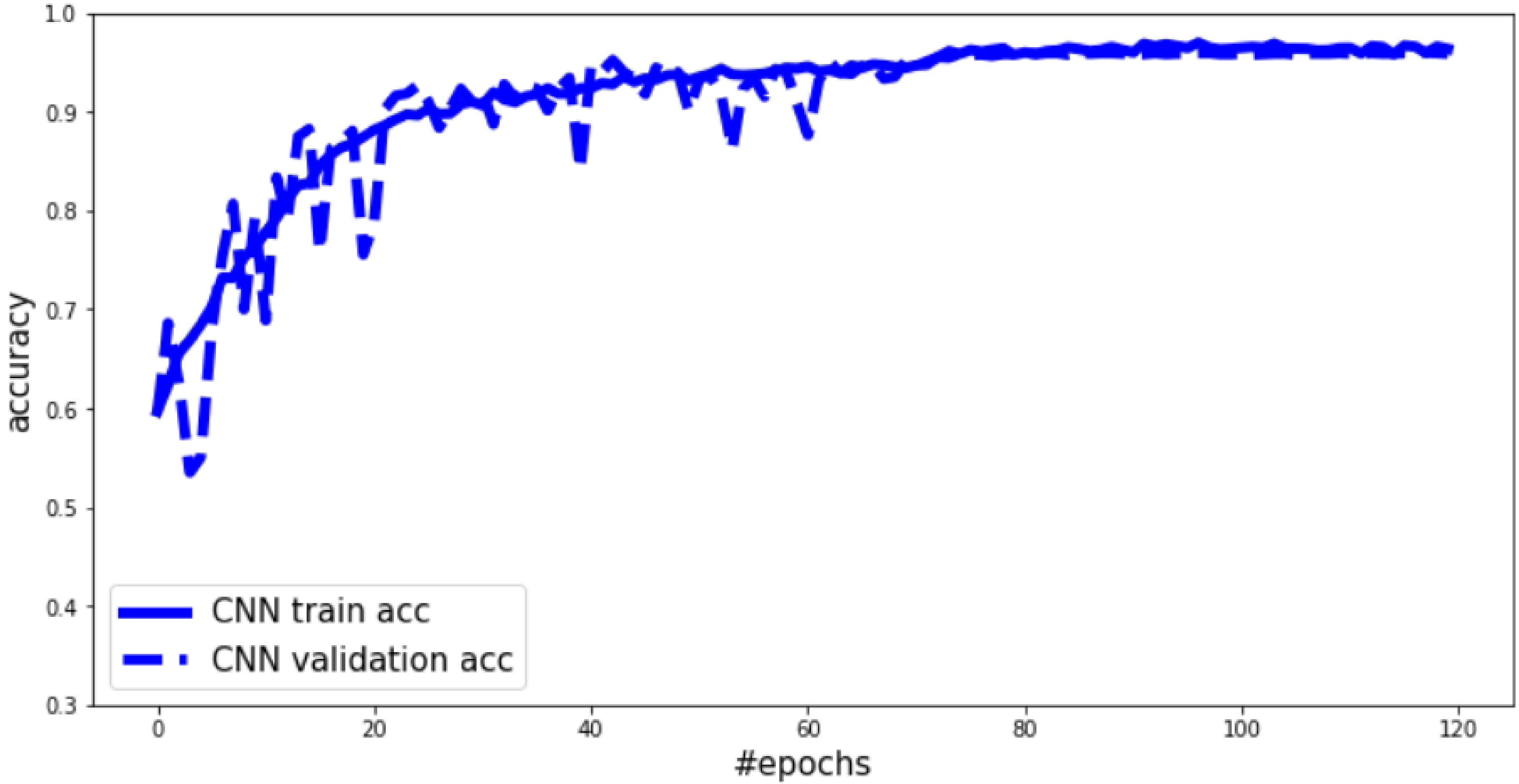


MŰEGYETEM 1782

Accuracy for different network architectures



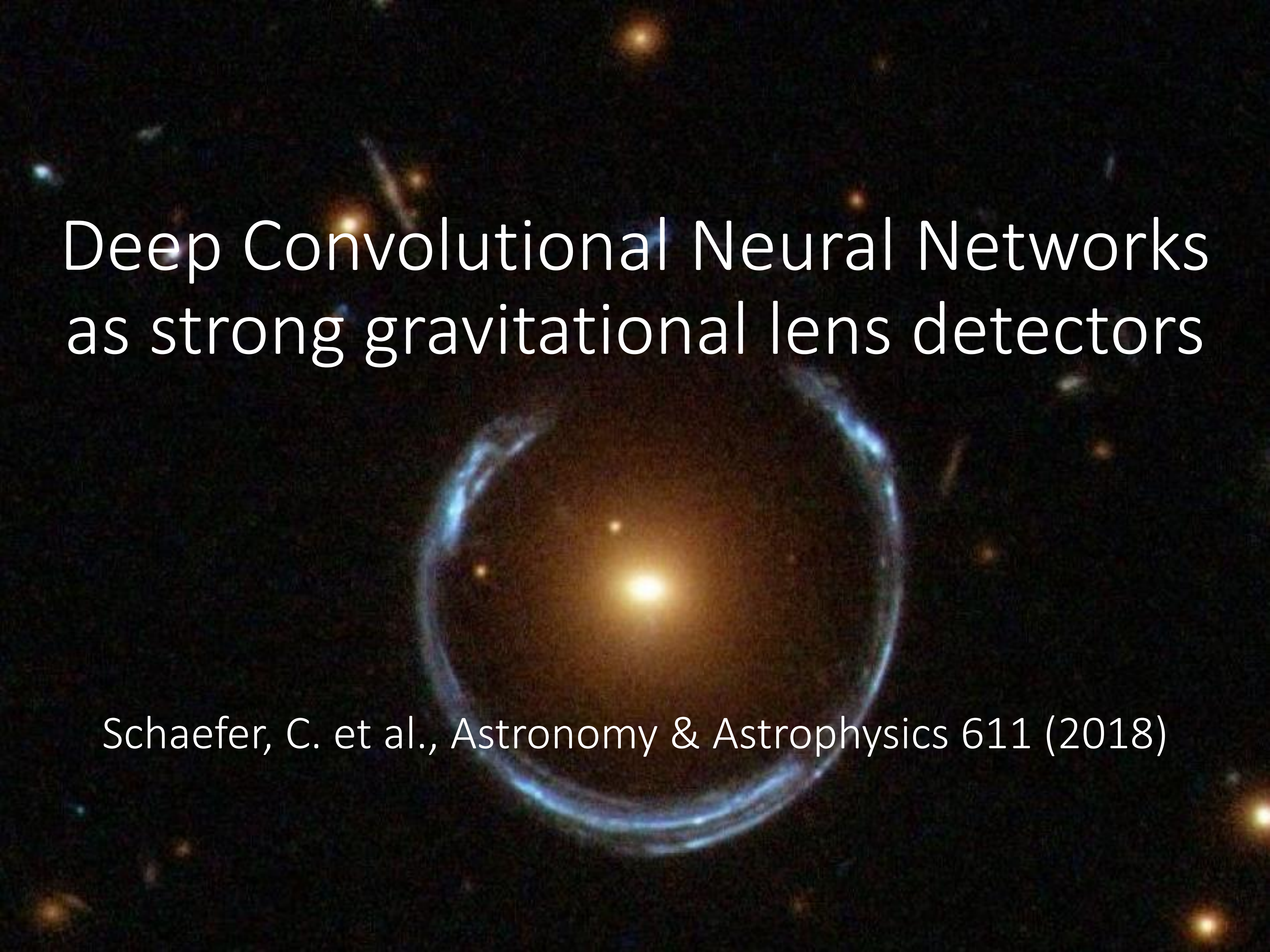
Accuracy for different network architectures



Winning network architecture *(96 % accuracy)*

```
cnn_model = Sequential()

cnn_model.add(Conv2D(32, (3, 3), activation = 'elu', input_shape=input_shape))
cnn_model.add(BatchNormalization())
cnn_model.add(MaxPooling2D((2, 2)))

cnn_model.add(Conv2D(32, (3, 3), activation = 'elu', input_shape=input_shape))
cnn_model.add(BatchNormalization())
cnn_model.add(MaxPooling2D((2, 2)))

cnn_model.add(Conv2D(64, (3, 3), activation = 'elu'))
cnn_model.add(BatchNormalization())
cnn_model.add(MaxPooling2D((2, 2)))

cnn_model.add(Conv2D(128, (3, 3), activation = 'elu'))
cnn_model.add(BatchNormalization())
cnn_model.add(MaxPooling2D((2, 2)))

cnn_model.add(Conv2D(128, (3, 3), activation = 'elu'))
cnn_model.add(BatchNormalization())
cnn_model.add(MaxPooling2D((2, 2)))


cnn_model.add(Flatten())
cnn_model.add(Dropout(0.5))  #Dropout for regularization
cnn_model.add(Dense(256, activation='relu'))
cnn_model.add(Dense(128, activation='relu'))
cnn_model.add(Dense(1, activation='sigmoid'))
cnn_model.compile(optimizer =  Adam() , loss = "binary_crossentropy", metrics=["accuracy"])
```

Deep Convolutional Neural Networks as strong gravitational lens detectors



Schaefer, C. et al., Astronomy & Astrophysics 611 (2018)

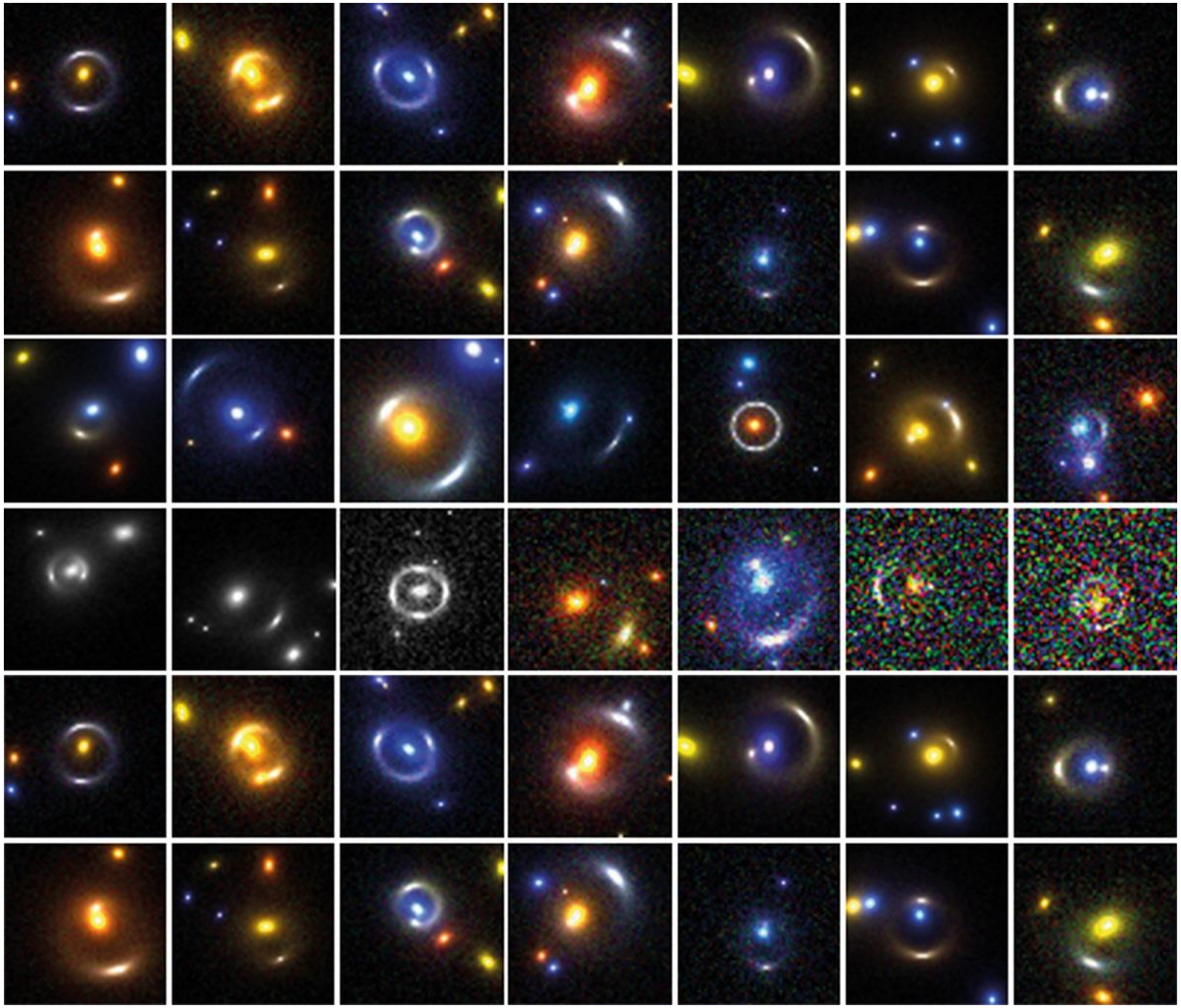
Gravitational lens

Matter that through the bending of space in its gravitational field alters the direction of light passing nearby.

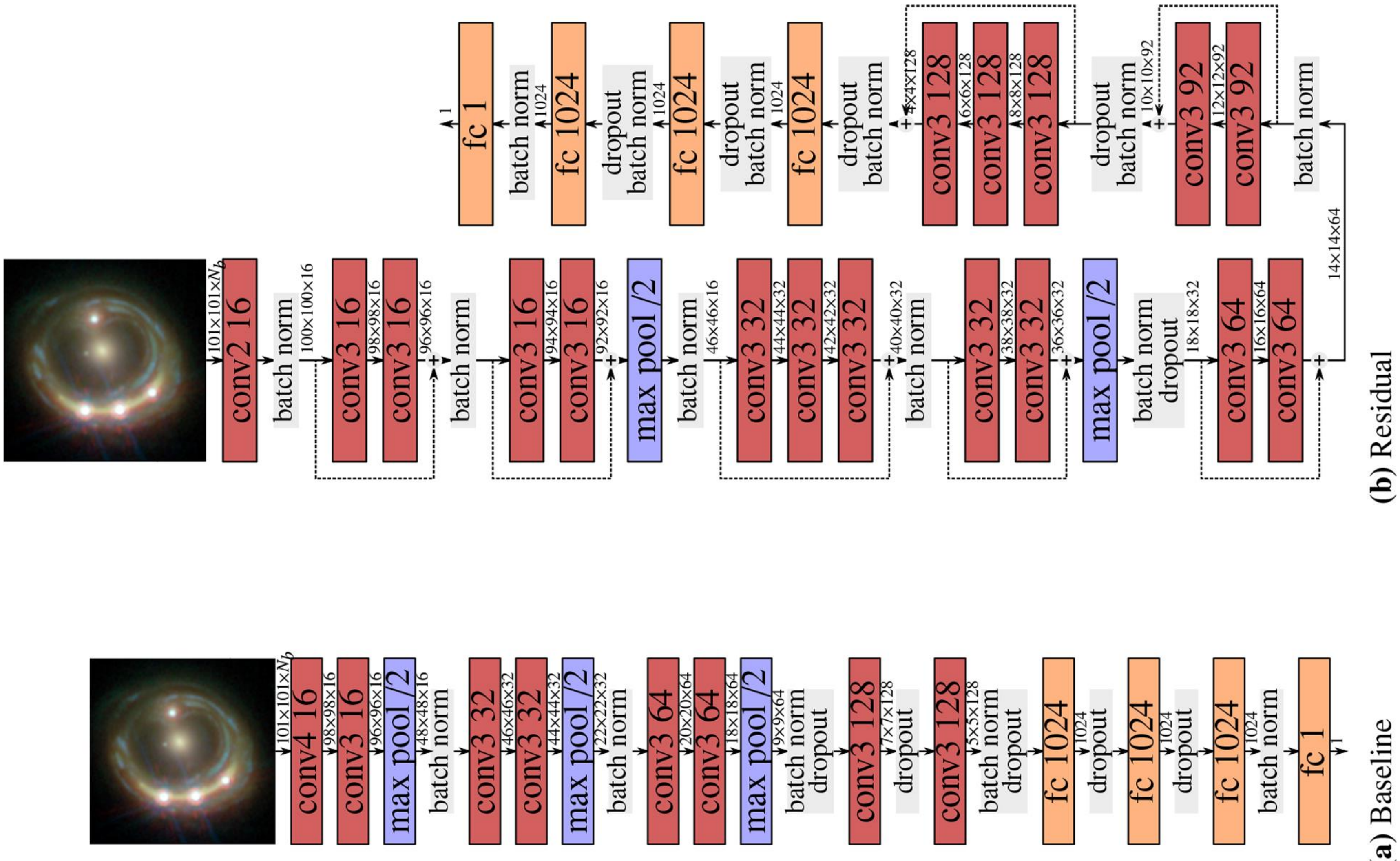
www.eso.org

A light source passes behind a gravitational lens

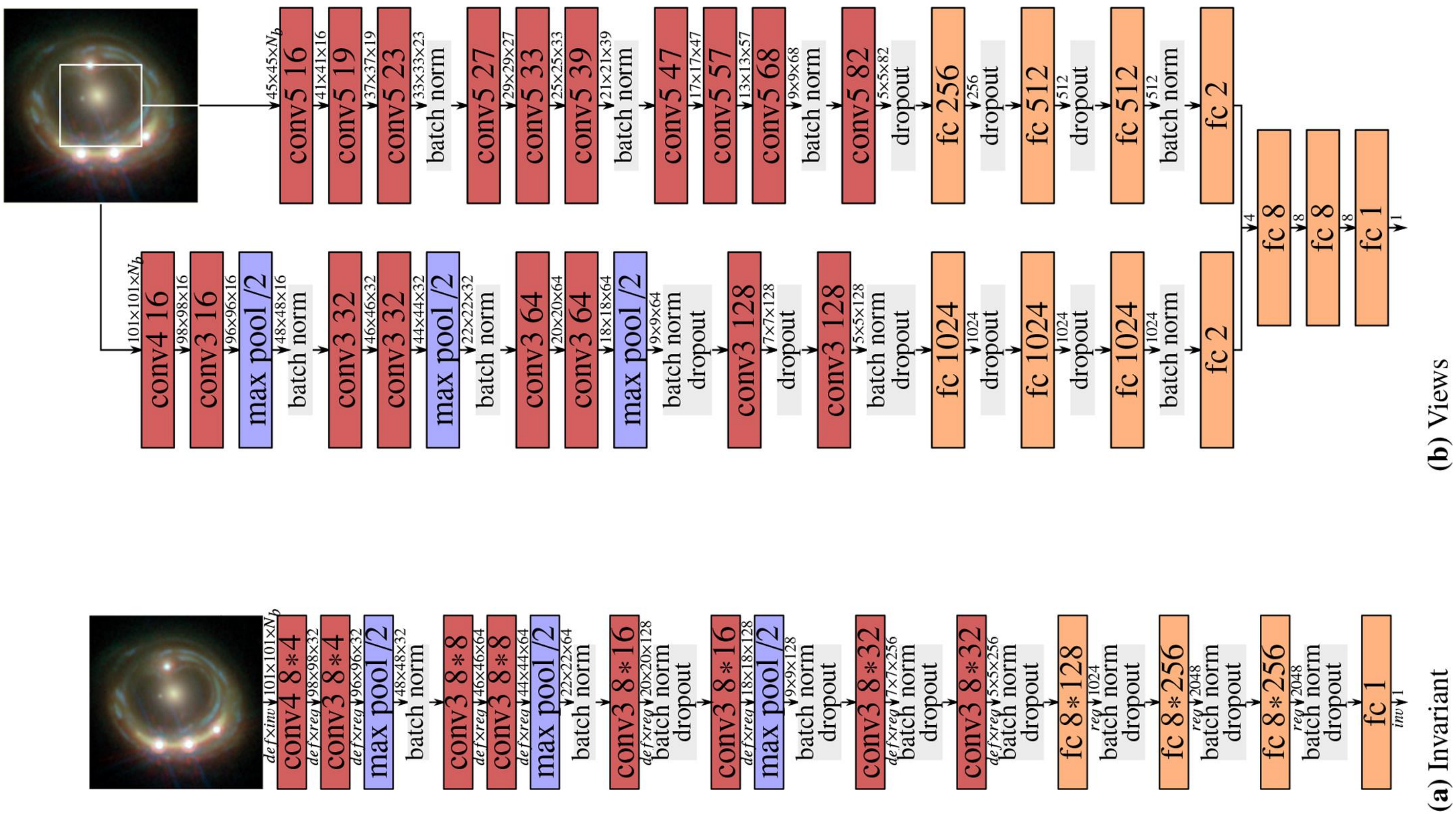
Strong Lensing challenge by the BLF



CNNs used for lens finding



CNNs used for lens finding



Possible pitfalls

- Black-box solutions
- Solving the wrong problem

(a) Husky classified as wolf (b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

Ribeiro MT et al. 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 Aug 13 (pp. 1135-1144). ACM.

Summary

- ↳ Automated feature extraction
- ↳ Solution of complex problems
- ↳ A complement to critical thinking and human expertise

My presentation is based on the following online article:
Neural networks, explained – written by Janelle Shane
https://physicsworld.com/a/neural-networks-explained/

Summary

- Automated feature extraction
- Solution of complex problems
- A complement to critical thinking and human expertise

My presentation is based on the following online article:
Neural networks, explained – written by Janelle Shane
https://physicsworld.com/a/neural-networks-explained/

Thank you for your attention!