Master's degree ICT Internet Multimedia Engineering

LAB 3: INCEPTION-V4 FOR CLASSIFICATION

Eleonora Cicciarella, Cesare Bidini
eleonora.cicciarella@phd.unipd.it
cesare.bidini@phd.unipd.it

University of Padova, IT

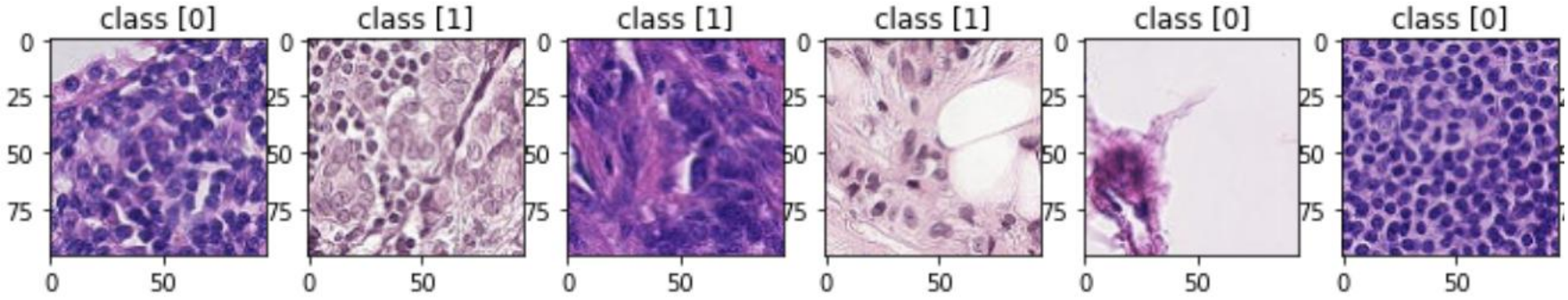
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO MATEMATICA

UNIVERSITÀ DEGLI STUDI DI PADOVA

Lab 3

- Inception-v4 for classification
  - TensorFlow/Keras
  - The challenge:
    - metastatic tissue identification
  - You will learn to:
    - implement the basic building blocks of Inception-v4
    - put together these building blocks to implement and train a state-of-the-art neural network for binary image classification



CNN milestones

- Improve performance on large-scale image recognition tasks
  - AlexNet (2012)
  - VGG (2014)
  - Inception v1 - v4 (2014-2016)
  - ResNet (2015)
  - Inception-ResNet (2016)
  - DenseNet (2017)
  - SqueezeNet (2016)
  - EfficientNet (2019)
  - VisionTransformer (ViT) (2020)

Inception

- **Deep CNNs** extract features from images hierarchically
  - early layers capture simple patterns
  - deeper layers learn increasingly abstract and complex representations
- **Motivation for larger networks**: increasing *depth* (more layers) or *width* (more units) can improve performance
- Drawbacks of very deep networks:
  - They are prone to overfitting (large # of parameters)
  - They require huge computational resources

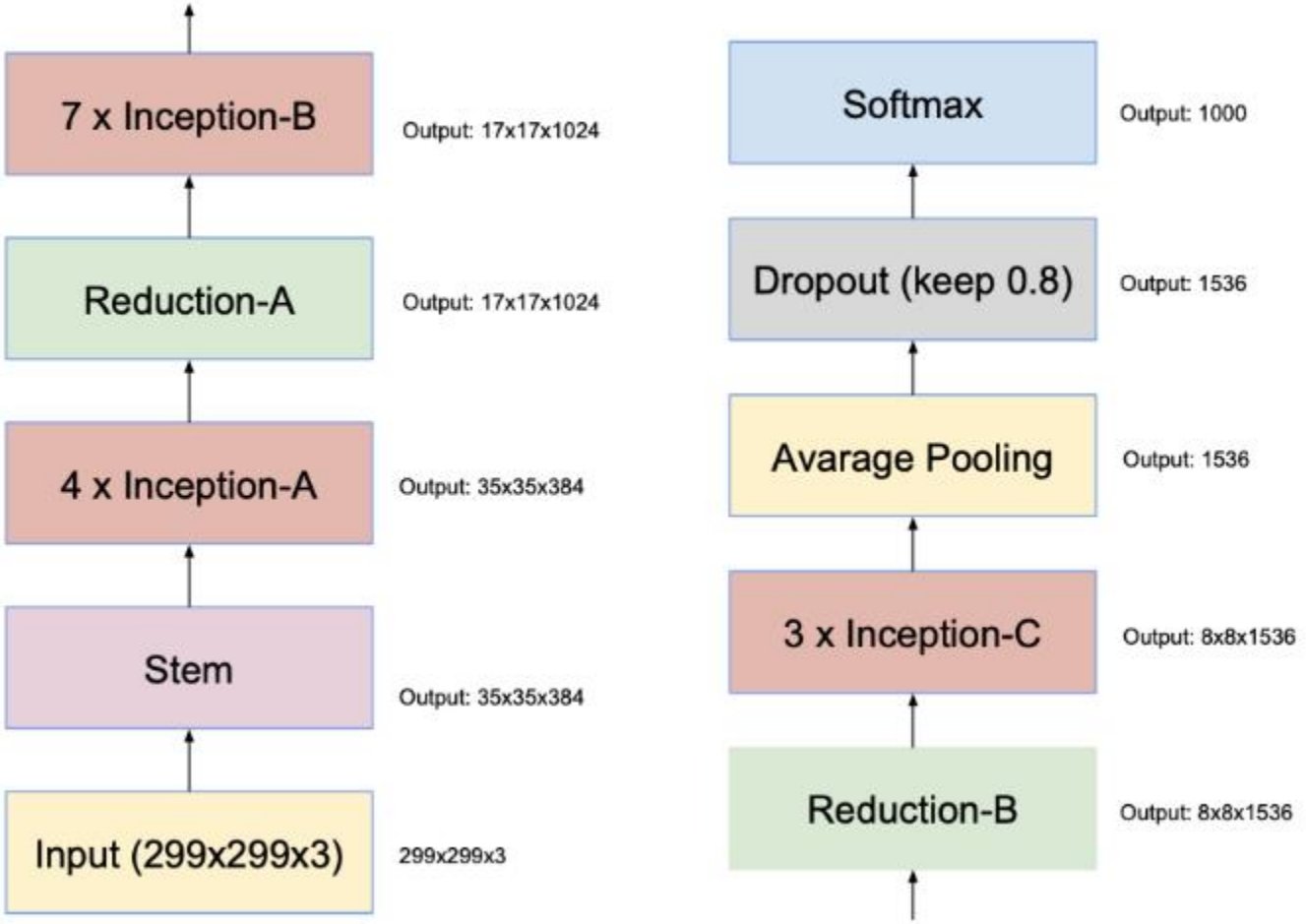
Inception

- Main idea of Inception architecture:
  - go wider (instead of only deeper)
  - each inception block applies **multiple convolutions in parallel** to learn features at different scales
  - outputs are **concatenated** and passed to the next layer for richer feature learning
- Inception-v4 deepens the network, but
  - uses many small convolutions (e.g., replace 5x5 with two 3x3)
  - applies dimensionality reduction with 1x1 convolutions

- Introduced for the first time by Google in 2014 (GoogLeNet): https://arxiv.org/pdf/1409.4842.pdf
- In this lab you will implement the **Inception-v4** network (2016) https://arxiv.org/pdf/1602.07261.pdf

Inception: three main blocks

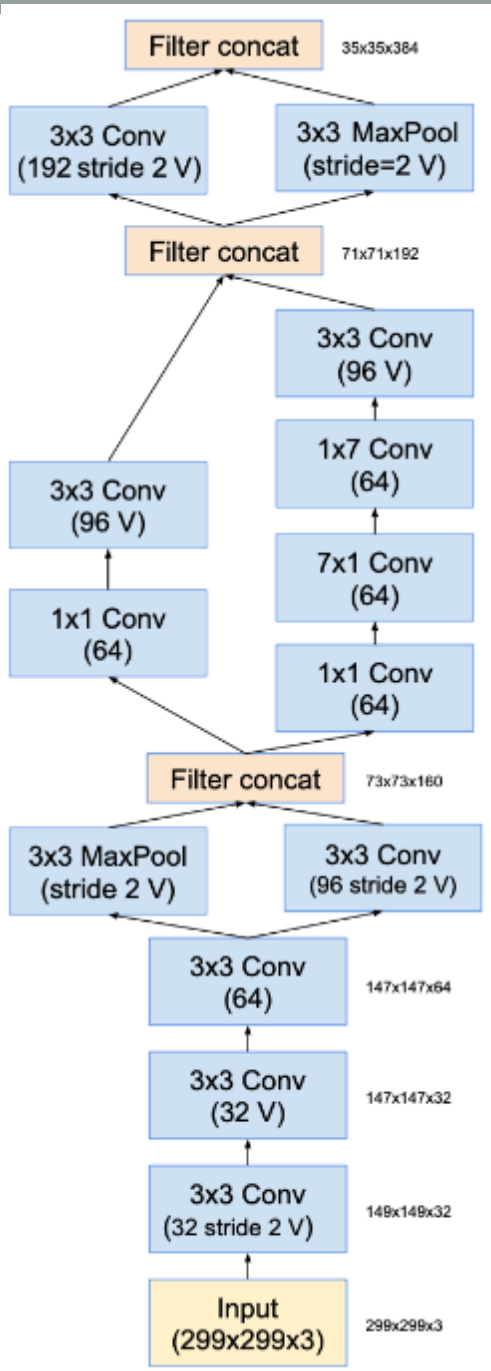
Stem block - Inception block - Reduction block



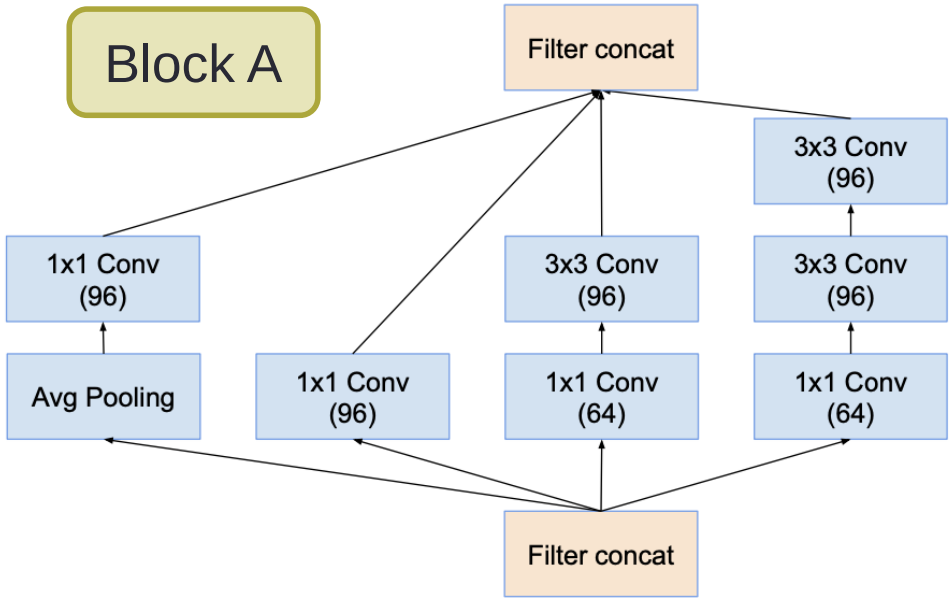
STEM block

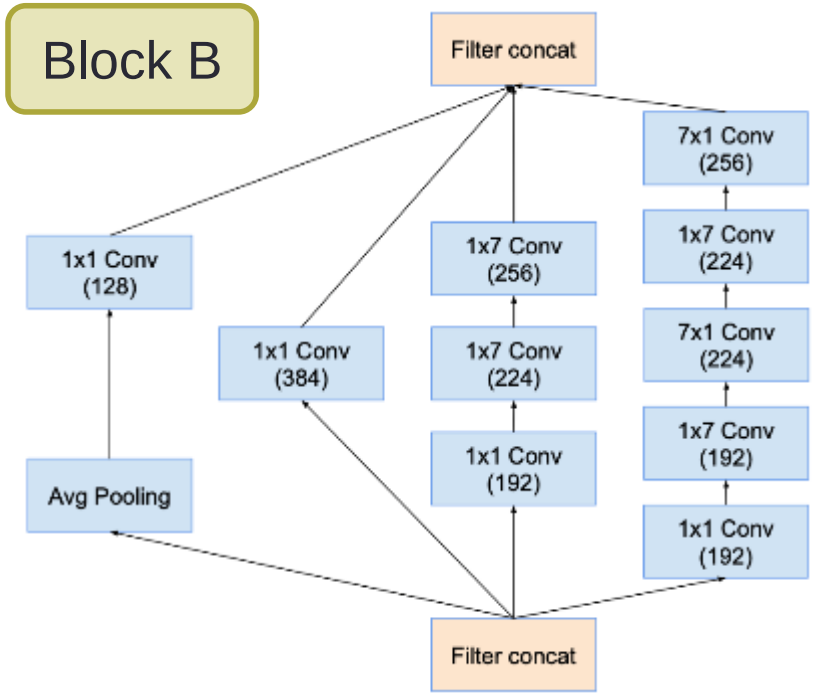
- Prepare data for deeper blocks
- Reduce image size (stride)
- Increase number of features maps

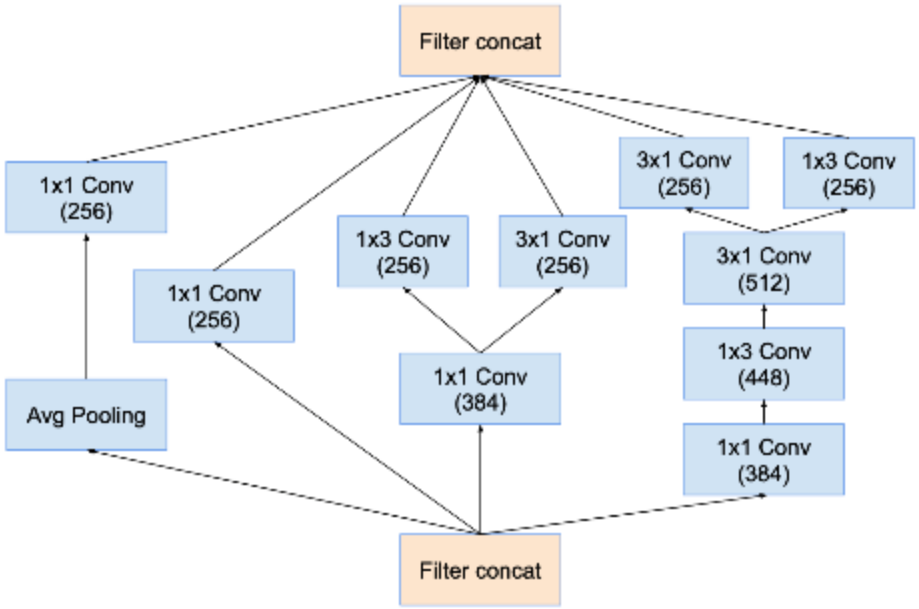
- From (299,299,3) to (35,35,384)



Inception blocks



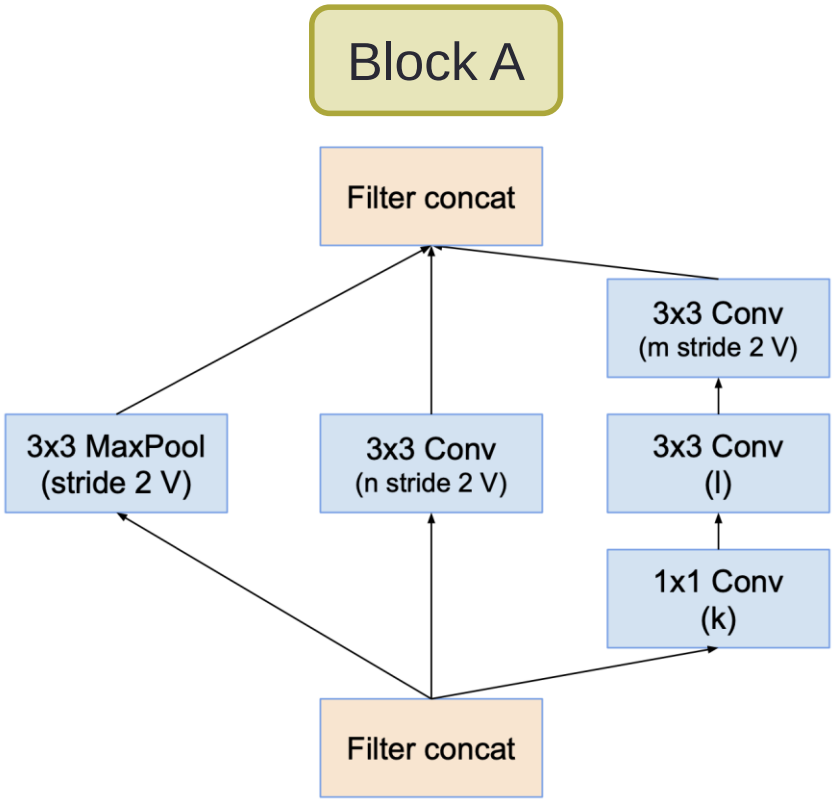


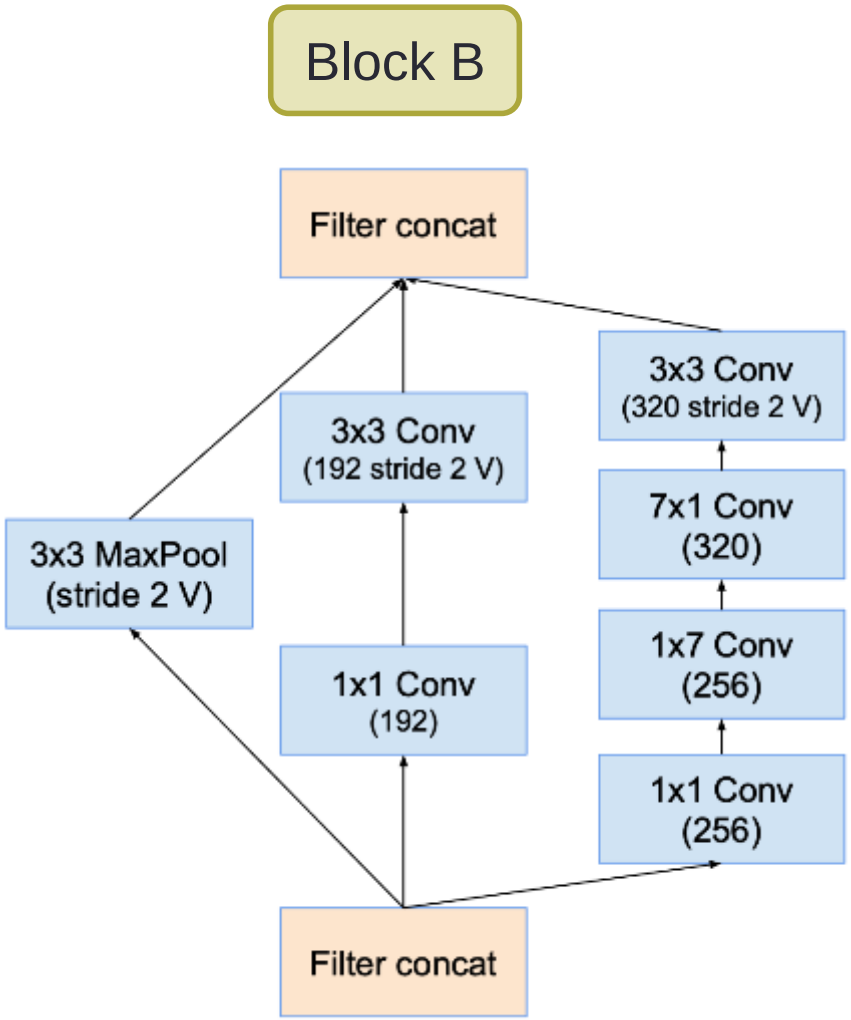


Block C

- Extract features at different spatial scales simultaneously

Reduction blocks

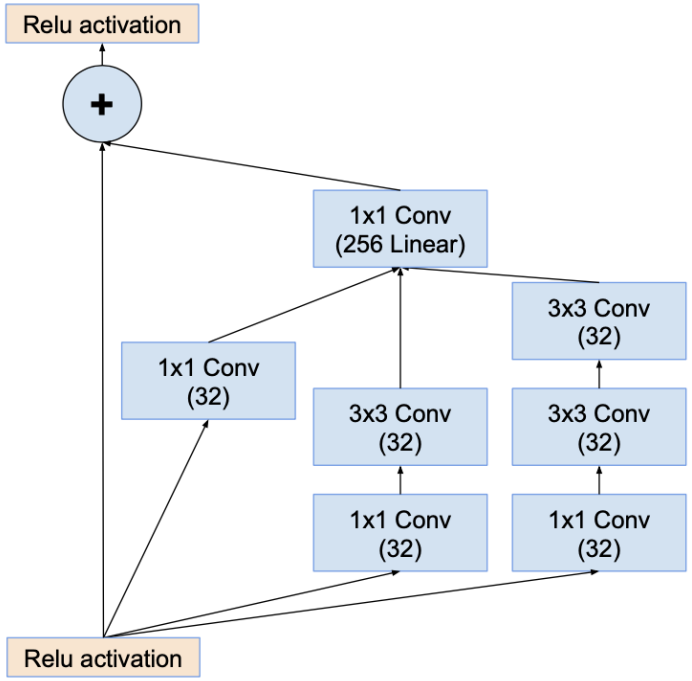




- Reduce spatial resolution while increasing depth

Inception-ResNet

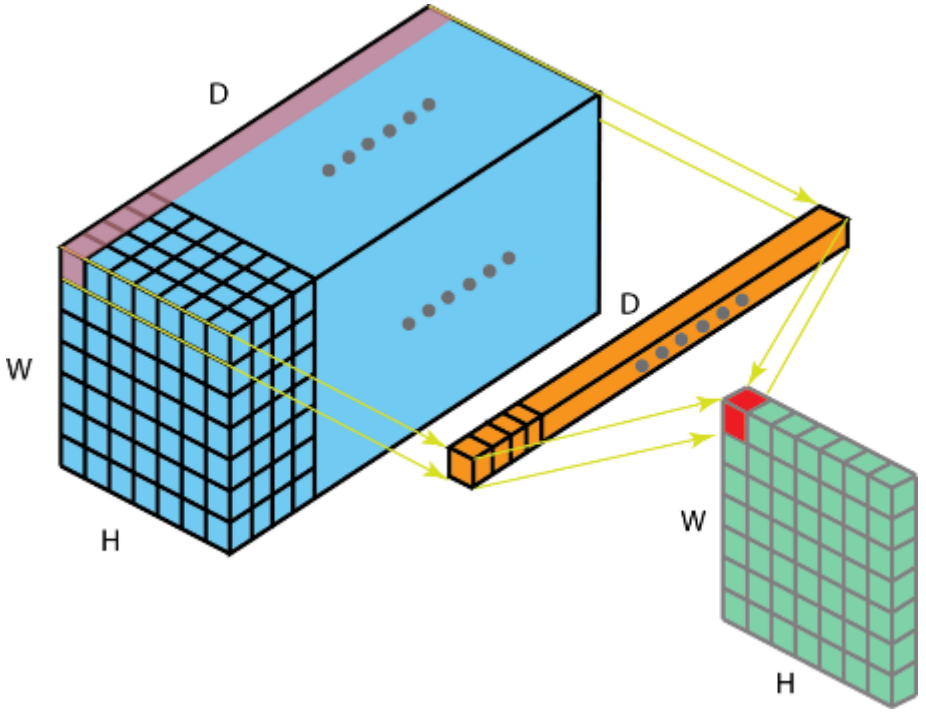
- Another architecture has been presented in https://arxiv.org/pdf/1602.07261.pdf combining the powerful of Inception and Residual networks



- you can try to experiment also with this structure by combining the two architectures you have seen in this class

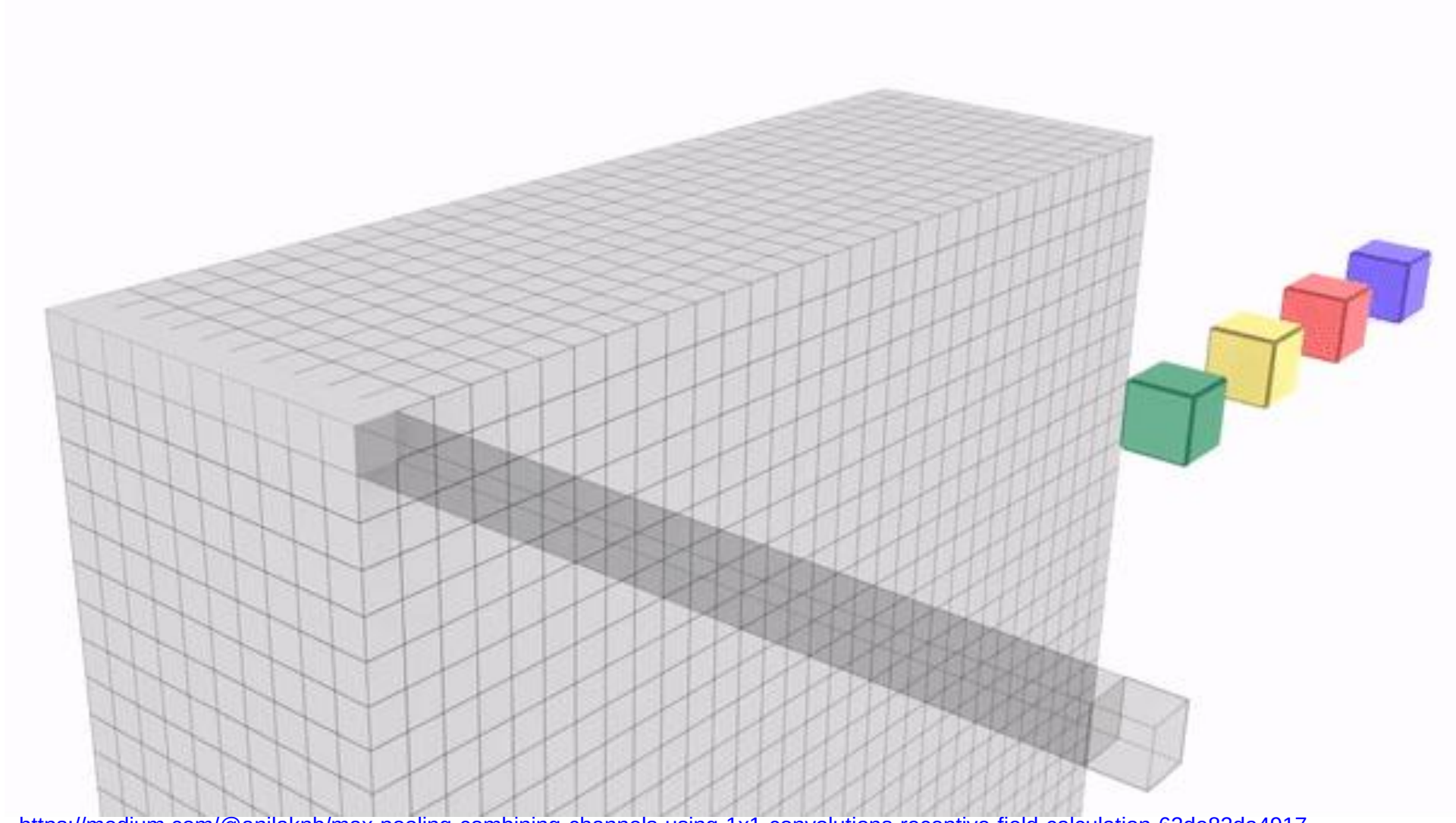
About 1x1 convolutions

- Used to reduce or increase the number of feature maps
- Preserve the input shape (padding is used)
- Actually…they do not use convolutions as a single input value is considered at a time



https://machinelearningmastery.com/introduction-to-1x1-convolutions-to-reduce-the-complexity-of-convolutional-neural-networks/

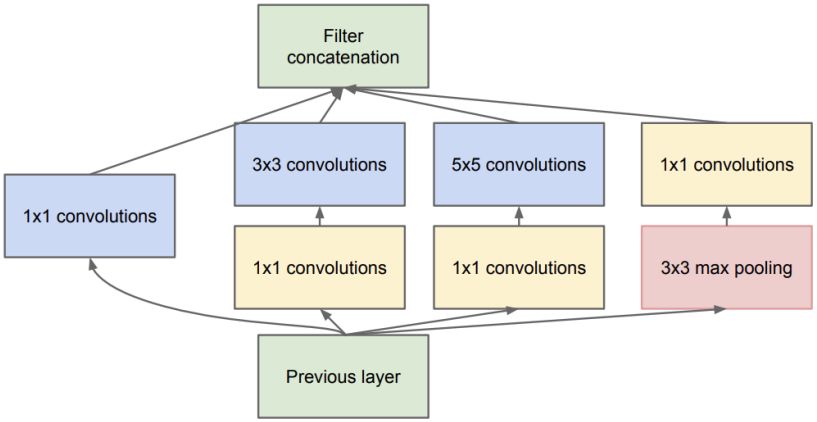
About 1x1 convolutions



https://medium.com/@anilaknb/max-pooling-combining-channels-using-1x1-convolutions-receptive-field-calculation-62de82da4917
https://medium.com/analytics-vidhya/talented-mr-1x1-comprehensive-look-at-1x1-convolution-in-deep-learning-f6b355825578#:~:text=By%20adding%201X1%20Conv%20layer,end%20up%20being%20more%20efficient.

1x1 conv. in Inception

- Why? Let's check the reasons from Google
  https://arxiv.org/pdf/1409.4842.pdf



(b) Inception module with dimension reductions

One big problem with the above modules, at least in this naïve form, is that even a modest number of 5×5 convolutions can be prohibitively expensive on top of a convolutional layer with a large number of filters. This problem becomes even more pronounced once pooling units are added to the mix: their number of output filters equals to the number of filters in the previous stage. The merging of the output of the pooling layer with the outputs of convolutional layers would lead to an inevitable increase in the number of outputs from stage to stage. Even while this architecture might cover the optimal sparse structure, it would do it very inefficiently, leading to a computational blow up within a few stages.

This leads to the second idea of the proposed architecture: judiciously applying dimension reductions and projections wherever the computational requirements would increase too much otherwise. This is based on the success of embeddings: even low dimensional embeddings might contain a lot of information about a relatively large image patch. However, embeddings represent information in a dense, compressed form and compressed information is harder to model. We would like to keep our representation sparse at most places (as required by the conditions of [2]) and compress the signals only whenever they have to be aggregated en masse. That is, 1×1 convolutions are used to compute reductions before the expensive 3×3 and 5×5 convolutions. Besides being used as reductions, they also include the use of rectified linear activation which makes them dual-purpose. The final result is depicted in Figure 2(b).

Lab 3 - Classification of metastatic tissue

- Inception-v4 was originally trained on **ImageNet** (299x299x3 images)
- In this lab, we will use a **different dataset** with smaller images: 96x96x3
- Hence, it is possible that some layers will have different values compared to the original architecture

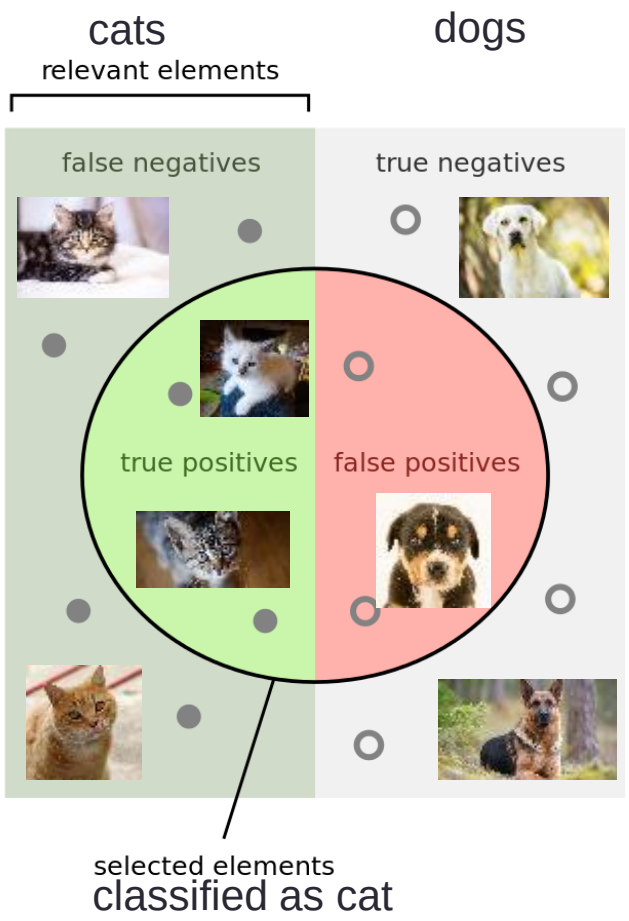
EVALUATION METRICS

Evaluation metrics

In a binary classification task

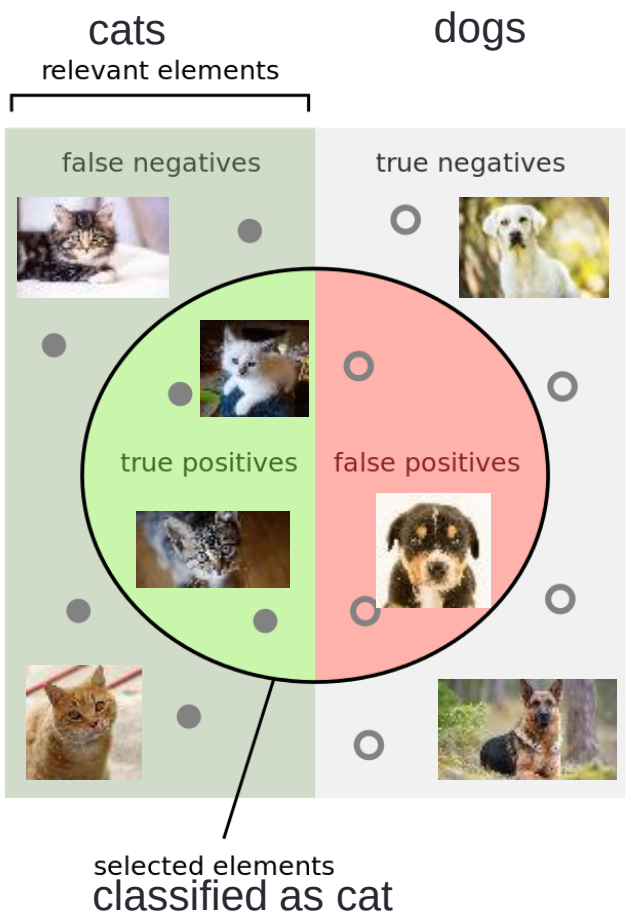
- Positives
  - no. of cases in which the algorithm says that an event is verified (inside the circle in the figure)
  - **TRUE**: positive cases that are correctly classified
  - **FALSE**: negatives that are classified as positives

- Negatives
  - no. of cases in which the algorithm says that an event is not verified (outside the circle in the figure)
  - **TRUE**: negatives that are correctly classified
  - **FALSE**: positives that are classified as negatives



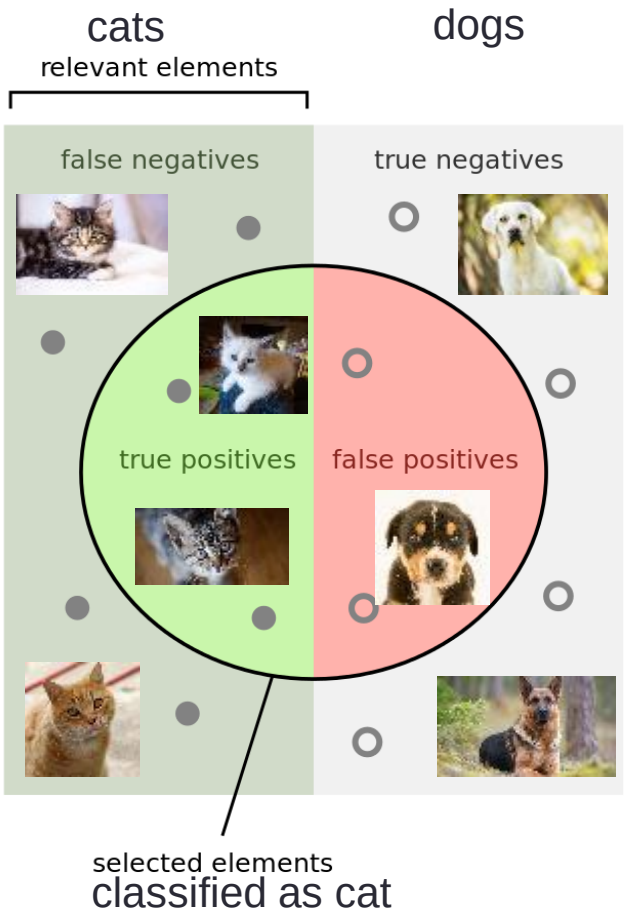
FPR and TPR

- False Positive Rate
  - False positives, divided by
  - Total of real negatives (False positives + **True negatives**)
- True Positive Rate
  - True positives, divided by
  - Total of real positives (True positives + **False negatives**)
- Receiving operating characteristic (ROC) curve
  - TPR – FPR pairs
  - varying the threshold used to infer the estimated classes (remember, binary classification problem)



Precision & recall

- Precision
  - True positives, divided by
  - Total cases identified as positives by the algorithm (True positives + False positives)
- Recall (= True Positive Rate)
  - True positives, divided by
  - Total of real positives (True positives + **False negatives**)
- Precision-recall curve (PRC)
  - recall – precision pairs
  - varying the threshold used to infer the estimated classes



How many selected items are relevant?

Precision =

How many relevant items are selected?

Recall =

F-measure

The performance of a binary classification algorithm is usually evaluated

- through the **F_1 score** (also called **F-score** or **F-measure**)
- is a measure of classification accuracy
- it considers both the precision and the recall

The F_1 score is the harmonic average of precision and recall

- F_1 score reaches its highest value at 1 (perfect precision and recall) and its smallest at 0

$$F_1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Weighted F-measure

For a multiclass classification problem

- F-measure has to be extended to multiple classes
- Class imbalance has to be accounted for
- The following equation is used:

$$F_1 = \sum_i 2w_i \times \frac{\text{precision}_i \times \text{recall}_i}{\text{precision}_i + \text{recall}_i}$$

where $w_i = n_i/N$

- n_i: number of samples of class i
- N: total number of samples

Master's degree ICT Internet Multimedia Engineering

LAB 3: INCEPTION-V4 FOR CLASSIFICATION

Eleonora Cicciarella, Cesare Bidini
eleonora.cicciarella@phd.unipd.it
cesare.bidini@phd.unipd.it

University of Padova, IT

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO MATEMATICA

UNIVERSITÀ DEGLI STUDI DI PADOVA