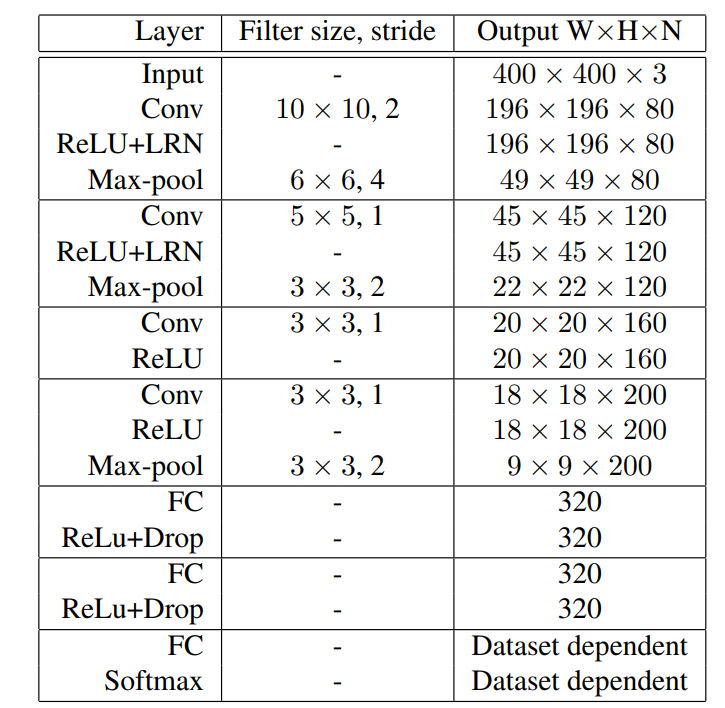
**Zelfstudie – Main project – 09-03-2022**

*Door: Myrthe Boone*

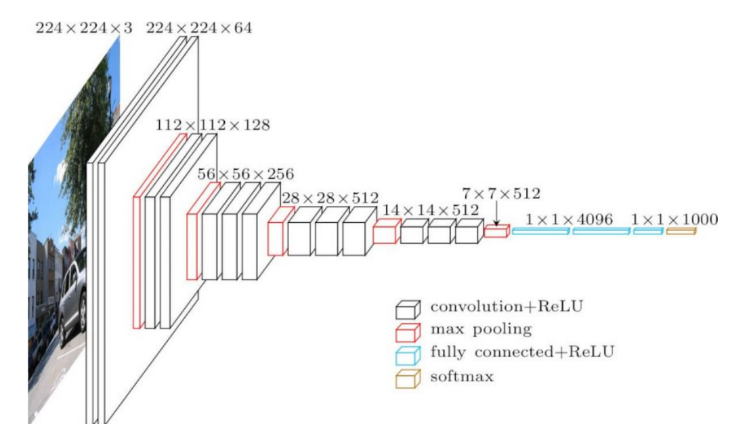
Opdracht : (literatuur) onderzoek over hoe een neural network kan worden gemaakt voor image recognition van histology patches

**Pretrained CNN / literature CNNs**

Overview of neural network used for image recognition of gliomas and Non-Small-Cell Lung Carcinoma (NSCLC) from the TCGA dataset [1]



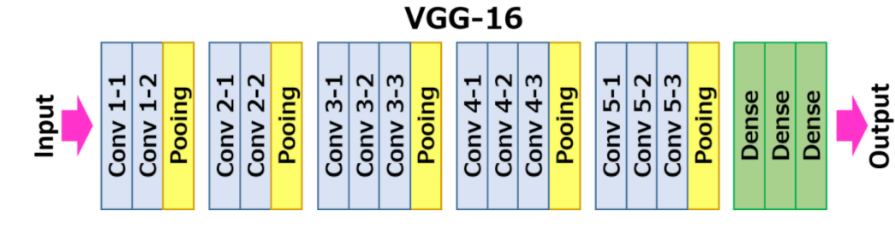
**Question:** If we use a pretrained CNN (and change some layers / output layer), it is a type of transfer learning right? But I thought they indicated in assignment 4 that we do not have to use transfer learning?

**Pretrained networks [2] (artikel komt uit augustus 2020)**

1. **VGG-16**

* Sequential in nature and uses a lot of filters
* 138 billion parameters, slower and much larger model to train than the others

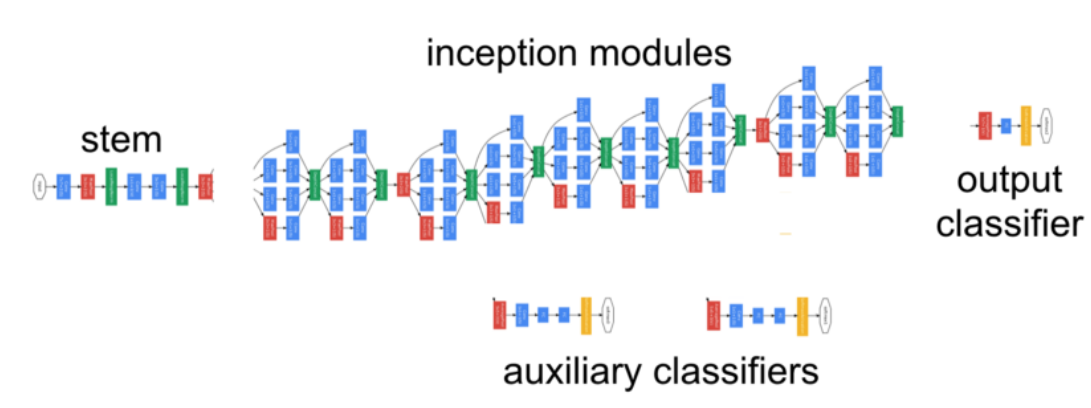
Architecture:

Can be directly imported from keras

Paper on VGG: <https://arxiv.org/pdf/1409.1556.pdf>

1. **GoogLeNet / Inception**

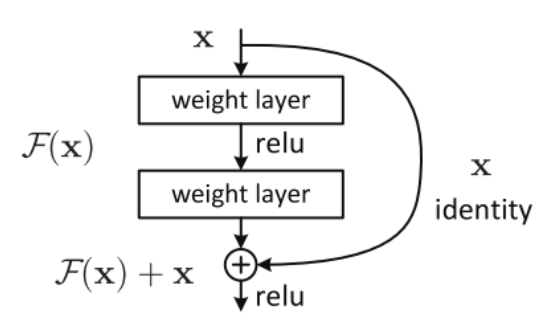
* Only 7 million parameters, smaller dan VGG and AlexNet
* Lower error rate
* Convolutions with different filter sizes on the input, performs Max Pooling, and concatenates the result for the next Inception module.
* The introduction of the 1 \* 1 convolution operation reduces the parameters drastically.
* 22 layers but reduction in number of parameters makes it a difficult model to beat
* Newest version:



Paper on inception: <https://arxiv.org/pdf/1512.00567v3.pdf>

1. **ResNet50**

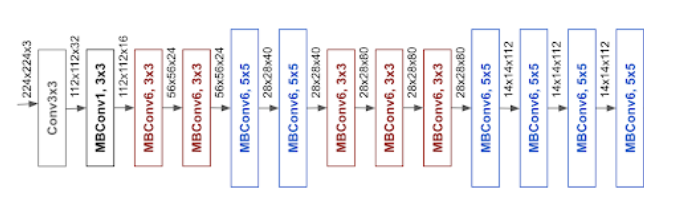
* Aims to tackle the vanishing gradient issue
* Main concept: after every two layers we bypass / skip the layer in between
  + Skipped connections are called ‘identity shortcut connections’ 🡪 uses the residual blocks
* Assumption / proposition: fitting a residual mapping much easier than fitting the actual mapping and thus apply it in all the layers
* Another notion of the authors: the more layers we stack, the model should not perform worse
  + Contrary to Inception and almost similar to VGG16 🡪 just ‘stacking layers’ on top of each other
* Top 5 error rate of 5%
* Series of architectures based on ResNet (not necessarily other versions but inspiration from ResNet) :



Paper on ResNet: <https://arxiv.org/pdf/1512.03385v1.pdf>

1. **EfficientNet (Google again)**

* New scaling method: compound scaling
  + Conventional approach: scale dimensions arbitrarily and by adding up more and more layers
  + New approach: scale dimensions by a fixed amount at the same time and do so uniformly 🡪 much better performance
* Scaling technique can be used for any CNN model
* Compound scaling formula
* Has to be installed separately via pip install in anaconda before it can be used
* Used in EfficientNetB0:



Paper on EfficientNet:

<https://arxiv.org/pdf/1905.11946.pdf>

**Additional tips and ideas ideas to improve existing neural networks / work further with model from assignment 3**

* Cross validation
* Image augmentation (if dataset is too small but what is too small?)
* Use fixed random seed [3]
* Start simple (e.g. save data augmentation and regularization for later)
  + “ E.g. if you are classifying images don’t be a hero and just copy paste a ResNet-50 for your first run. You’re allowed to do something more custom later and beat this.” [3]
  + So good idea to start with ResNet 😊
* Use Adam optimization algorithm (safe to start with, method for stochastic optimization)
* Only when we have a good understanding of how the network works, and which optimization algorithm we want to use we can go further with the next steps
  + Data augmentation
  + Regularization
  + Drop out layers
  + Use pretrained model
  + Decrease batch size
  + Random grid search + hyper parameter optimization

**Question**: is an exisiting CNN architecture (ResNet e.g. ) also a pretrained network or is that something different?

**Bibliography**

[1]

<https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Hou_Patch-Based_Convolutional_Neural_CVPR_2016_paper.pdf>

[2] <https://www.analyticsvidhya.com/blog/2020/08/top-4-pre-trained-models-for-image-classification-with-python-code/>

[3] <http://karpathy.github.io/2019/04/25/recipe/>