References

**W​eek 1:**

* [The Sequential model](https://www.tensorflow.org/guide/keras/sequential_model) (TensorFlow Documentation)
* [The Functional API](https://www.tensorflow.org/guide/keras/functional) (TensorFlow Documentation)

**W​eek 2:**

* [Deep Residual Learning for Image Recognition](https://arxiv.org/abs/1512.03385) (He, Zhang, Ren & Sun, 2015)
* [d​eep-learning-models/resnet50.py/](https://github.com/fchollet/deep-learning-models/blob/master/resnet50.py) (GitHub: fchollet)
* [MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications](https://arxiv.org/abs/1704.04861) (Howard, Zhu, Chen, Kalenichenko, Wang, Weyand, Andreetto, ​& Adam, 2017)
* [MobileNetV2: Inverted Residuals and Linear Bottlenecks](https://arxiv.org/abs/1801.04381) (Sandler, Howard, Zhu, Zhmoginov &Chen, 2018)
* [EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks](https://arxiv.org/abs/1905.11946) (Tan & Le, 2019)

**W​eek 3:**

* [You Only Look Once: Unified, Real-Time Object Detection](https://arxiv.org/abs/1506.02640) (Redmon, Divvala, Girshick & Farhadi, 2015)
* [YOLO9000: Better, Faster, Stronger](https://arxiv.org/abs/1612.08242) (Redmon & Farhadi, 2016)
* [YAD2K](https://github.com/allanzelener/YAD2K) (GitHub: allanzelener)
* [YOLO: Real-Time Object Detection](https://pjreddie.com/darknet/yolo/)
* [Fully Convolutional Architectures for Multi-Class Segmentation in Chest Radiographs](https://arxiv.org/abs/1701.08816) (Novikov, Lenis, Major, Hladůvka, Wimmer & Bühler, 2017)
* [Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks](https://arxiv.org/abs/1705.03820) (Dong, Yang, Liu, Mo & Guo, 2017)
* [U-Net: Convolutional Networks for Biomedical Image Segmentation](https://arxiv.org/abs/1505.04597) (Ronneberger, Fischer & Brox, 2015)

**W​eek 4:**

* [FaceNet: A Unified Embedding for Face Recognition and Clustering](https://arxiv.org/pdf/1503.03832.pdf) (Schroff, Kalenichenko & Philbin, 2015)
* [DeepFace: Closing the Gap to Human-Level Performance in Face Verification](https://research.fb.com/wp-content/uploads/2016/11/deepface-closing-the-gap-to-human-level-performance-in-face-verification.pdf) (Taigman, Yang, Ranzato & Wolf)
* [f​acenet](https://github.com/davidsandberg/facenet) (GitHub: davidsandberg)
* [How to Develop a Face Recognition System Using FaceNet in Keras](https://machinelearningmastery.com/how-to-develop-a-face-recognition-system-using-facenet-in-keras-and-an-svm-classifier/) (Jason Brownlee, 2019)
* [k​eras-facenet/notebook/tf\_to\_keras.ipynb](https://github.com/nyoki-mtl/keras-facenet/blob/master/notebook/tf_to_keras.ipynb) (GitHub: nyoki-mtl)
* [A Neural Algorithm of Artistic Style](https://arxiv.org/abs/1508.06576) (Gatys, Ecker & Bethge, 2015)
* [Convolutional neural networks for artistic style transfer](https://harishnarayanan.org/writing/artistic-style-transfer/)
* [TensorFlow Implementation of "A Neural Algorithm of Artistic Style"](http://www.chioka.in/tensorflow-implementation-neural-algorithm-of-artistic-style)
* [V​ery Deep Convolutional Networks For Large-Scale Image Recognition](https://arxiv.org/pdf/1409.1556.pdf) (Simonyan & Zisserman, 2015)
* [Pretrained models](https://www.vlfeat.org/matconvnet/pretrained/) (MatConvNet)