**Basics of Image Preprocessing -**

<https://www.mygreatlearning.com/blog/introduction-to-image-pre-processing/>

<https://colab.research.google.com/drive/1A9BUqXmZballXQGMSTz1EKSNzbWfdxpA#scrollTo=Al41JtXyZh9F>

**Smaller v/s Higher number of filters -**

<https://icecreamlabs.com/2018/08/19/3x3-convolution-filters%e2%80%8a-%e2%80%8aa-popular-choice/>

**Odd size filters -**

<https://towardsdatascience.com/deciding-optimal-filter-size-for-cnns-d6f7b56f9363>

**Edge Detection -**

<https://medium.com/perspectivesondatascience/preprocessing-with-computer-vision-part-v-edge-detection-1b51e37d5d3f>

<https://www.analytixlabs.co.in/blog/what-is-image-segmentation/>

**Morphological Operation -**

<https://medium.com/perspectivesondatascience/preprocessing-with-computer-vision-part-vii-morphological-operations-ca850a701ea8>

**Contour Detection-**

<https://medium.com/perspectivesondatascience/preprocessing-with-computer-vision-part-vi-contour-detection-68a6fb2942e0>

**Image Annotation -**

<https://github.com/MeghanaKshirsagar/Perspectives-on-Data-science/blob/main/Preprocessing/Image_Annotation.ipynb>

<https://medium.com/perspectivesondatascience/preprocessing-with-computer-vision-part-viii-image-annotation-227254b1c1bf>

**Affine Transformations -**

<https://towardsdatascience.com/image-geometric-transformation-in-numpy-and-opencv-936f5cd1d315#:~:text=Geometric%20transformation%20is%20an%20essential,image%20to%20another%20image%20plane>

<https://github.com/darylclimb/image_affine_transform/blob/master/transformation.ipynb>

**Omniglot Dataset -**

<https://www.geeksforgeeks.org/omniglot-classification-task/#:~:text=Omniglot%20Dataset%3A%20It%20is%20a,task%20from%20just%20one%20example>

**Kernal/Filter Size and odd size filters -**

<https://medium.com/analytics-vidhya/how-to-choose-the-size-of-the-convolution-filter-or-kernel-size-for-cnn-86a55a1e2d15>

<https://towardsdatascience.com/deciding-optimal-filter-size-for-cnns-d6f7b56f9363#:~:text=The%20explanation%20is%20as%20follows,using%20an%20even%20sized%20kernel>.

<https://towardsdatascience.com/visualizing-how-filters-work-in-convolutional-neural-networks-cnns-7383bd84ad2c>

<https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/>

**CNN Filters -**

<https://towardsdatascience.com/understanding-parameter-sharing-or-weights-replication-within-convolutional-neural-networks-cc26db7b645a>

<https://deeplizard.com/learn/video/qSTv_m-KFk0>

**Output Image Size Formula -**

<https://iq.opengenus.org/output-size-of-convolution/#:~:text=Machine%20Learning%20(ML)%20cnn&text=In%20short%2C%20the%20answer%20is,%2F%20(stride%20width)%20%2B%201>

(W−K+2P)/S]+1

* W is the input volume - in your case 128
* K is the Kernel size - in your case 5
* P is the padding - in your case 0 i believe
* S is the stride - which you have not provided.

So, we input into the formula:

Output\_Shape = (128-5+0)/1+1

Output\_Shape = (124,124,40)

(7-3+0)/1 + 1 = 5

(5-3+0)/1 + 1 = 3

**Alexbet/Resnet CNN Architeture**

<https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96>

<https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/>

**Feature maps -**

<https://medium.com/dataseries/visualizing-the-feature-maps-and-filters-by-convolutional-neural-networks-e1462340518e>

<https://towardsdatascience.com/convolutional-neural-network-feature-map-and-filter-visualization-f75012a5a49c#:~:text=Feature%20maps%20are%20generated%20by,Convolutional%20layers%20in%20the%20model>

**Padding and Stride -**

<https://deeplizard.com/learn/video/qSTv_m-KFk0>

<https://www.analyticsvidhya.com/blog/2018/12/guide-convolutional-neural-network-cnn/>

<https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480>

**Coding -**

[Computer Vision with OpenCV](https://www.youtube.com/playlist?list=PLfne_5LgR4wFe9botq7n5NjaSJQmRQK7O)

**SSD-**

[Single Shot Detector | SSD | Object Detection Using SSD](https://www.youtube.com/watch?v=NUEim5bF0_0)

**Semantic Segmentation & U-Net -**

<https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47>

**Transposed Convolution -**

<https://naokishibuya.medium.com/up-sampling-with-transposed-convolution-9ae4f2df52d0>

**One/Few Shot Learning Siamese Network -**

<https://medium.com/swlh/one-shot-learning-with-siamese-network-1c7404c35fda>

<https://towardsdatascience.com/one-shot-learning-with-siamese-networks-using-keras-17f34e75bb3d>

<https://github.com/wangshusen/DeepLearning>

**Parameter Calculation -**

<https://towardsdatascience.com/how-to-calculate-the-number-of-parameters-in-keras-models-710683dae0ca>

**Important Notebooks -**

<https://colab.research.google.com/drive/1A9BUqXmZballXQGMSTz1EKSNzbWfdxpA?usp=sharing>

<https://colab.research.google.com/drive/1y42BoqT51M4NCsCgWD4mNkysnQzT1SZ6?usp=sharing>

**UNET -**

<https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47>

**One-shot Learning -**

<https://towardsdatascience.com/one-shot-learning-with-siamese-networks-using-keras-17f34e75bb3d>

**Object Detection -**

<https://towardsdatascience.com/computer-vision-instance-segmentation-with-mask-r-cnn-7983502fcad1>

<https://towardsdatascience.com/computer-vision-a-journey-from-cnn-to-mask-r-cnn-and-yolo-1d141eba6e04>

**Alexnet -**

<https://towardsdatascience.com/what-alexnet-brought-to-the-world-of-deep-learning-46c7974b46fc>

<https://medium.com/x8-the-ai-community/explaining-alexnet-convolutional-neural-network-854df45613aa>

<https://medium.com/coinmonks/paper-review-of-alexnet-caffenet-winner-in-ilsvrc-2012-image-classification-b93598314160>

**RELU-**

<https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7>

**Parameter Calculation -**

<https://towardsdatascience.com/understanding-and-calculating-the-number-of-parameters-in-convolution-neural-networks-cnns-fc88790d530d>

<https://stackoverflow.com/questions/42786717/how-to-calculate-the-number-of-parameters-for-convolutional-neural-network?source=post_page-----fc88790d530d>--------------------------------

<https://medium.com/@iamvarman/how-to-calculate-the-number-of-parameters-in-the-cnn-5bd55364d7ca#:~:text=To%20calculate%20it%2C%20we%20have,3%E2%80%931))%20%3D%2048>.

**Convolution Types -**

<https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610#:~:text=In%201D%20CNN%2C%20kernel%20moves,kernel%20moves%20in%202%20directions>.

<https://wandb.ai/ayush-thakur/dl-question-bank/reports/Intuitive-understanding-of-1D-2D-and-3D-convolutions-in-convolutional-neural-networks---VmlldzoxOTk2MDA>

**Semantic Segmentation -**

<https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47>

**Keras Code -**

<https://github.com/hlamba28/UNET-TGS/blob/master/TGS%20UNET.ipynb>

**Transposed Convolution -**

<https://naokishibuya.medium.com/up-sampling-with-transposed-convolution-9ae4f2df52d0>

**Code -**

<https://github.com/naokishibuya/deep-learning/blob/master/python/transposed_convolution.ipynb>

**Other Materials -**

<https://beyondminds.ai/blog/a-simple-guide-to-semantic-segmentation/>

**Valid v/s Same Padding -**

<https://stackoverflow.com/questions/37674306/what-is-the-difference-between-same-and-valid-padding-in-tf-nn-max-pool-of-t>

**MobileNets -**

<https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470>