**Apuntes doctorado**

**Notas:**

* Ordenador para computar Airsim: GigaByte
* Crear un prototipo 3D para Airsim
* <https://github.com/Microsoft/AirSim/wiki/hexacopter>
* <https://github.com/Microsoft/AirSim/blob/master/docs/settings.md>

**Timeline**

* Compilar script en Visual Basic con Airsim
* Saberlo utilizar tb en Linux
* Crear la red neuronal de Data+Image CRNN
* Generar Ground truth
* Crear vehículo personalizado para Airsim 3D + Physics
* Explorar la posibilidad de crear infraestructura de hardware
* Explorar arquitecturas de control
* Explorar aplicaciones de Amazon AWS

Problema NCHW

* <https://stackoverflow.com/questions/37689423/convert-between-nhwc-and-nchw-in-tensorflow>

Installing torch

**pip install torch torchvision**

<https://github.com/microsoft/AirSim/blob/master/PythonClient/car/DQNcar.py>

<https://rosenfelder.ai/multi-input-neural-network-pytorch/>

<https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>

<https://bdd-data.berkeley.edu/portal.html#download>

**The Blackbird UAV dataset**

<https://github.com/mit-fast/Blackbird-Dataset>

**Residual block**

BatchNormalization:

Batch normalization allows each layer of a network to learn by itself a little bit more independently of other layers.

* We can use higher learning rates because batch normalization makes sure that there’s no activation that’s gone really high or really low. And by that, things that previously couldn’t get to train, it will start to train.
* It reduces overfitting because it has a slight regularization effects. Similar to dropout, it adds some noise to each hidden layer’s activations. Therefore, if we use batch normalization, we will use less dropout, which is a good thing because we are not going to lose a lot of information. However, we should not depend only on batch normalization for regularization; we should better use it together with dropout.

<https://towardsdatascience.com/batch-normalization-in-neural-networks-1ac91516821c>

<https://towardsdatascience.com/residual-network-implementing-resnet-a7da63c7b278>

<https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py#L39-L51>