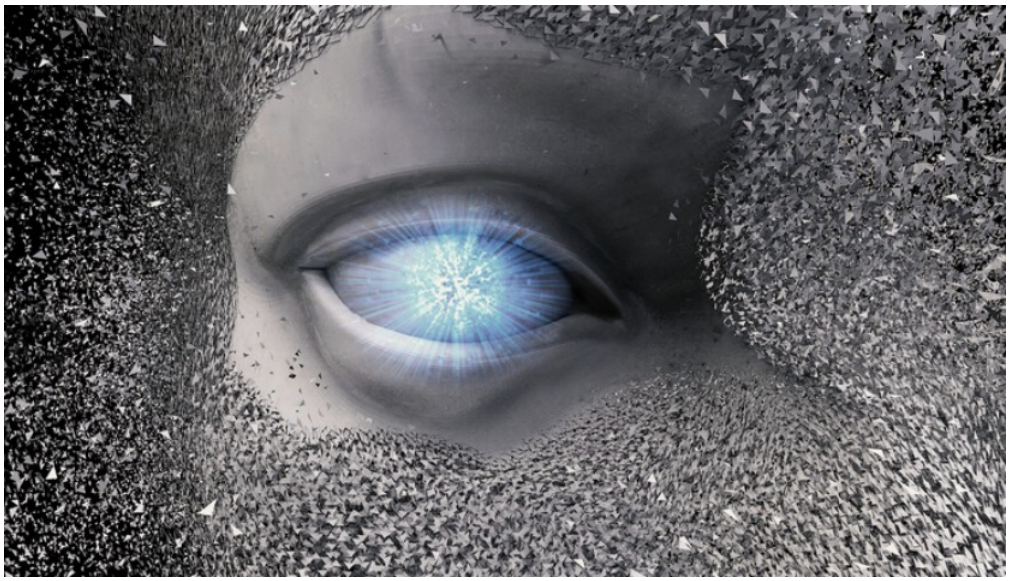
Deep Learning and Computer Vision A-Z™

SuperDataScience Team



Abstract

This course will help give you the core concepts addressing modern compter vision practices and techniques With this brand new course you will not only learn how the most popular computer vision methods work, but you will also learn to apply them in practice! We also use OpenCV and PyTorch as two prominent libraries that are highly utilized in Computer Vision, AI and Deep Learning research, and it's built with Python. It's recommended to view the documentation and readings as you progress that can be found here: `https://www.superdatascience.com/computer-vision/`

First, we would like to thank all of the students who have contributed to the discussions and Q&A sections. It's always a great idea and method of enhancing the materials to be able to discuss key concepts. In the following PDF you will see some of the main questions that we see frequently asked. If you don't see the answer that you are looking for please post in the Q&A in the course. In addition, the following section has some of the main setup and general questions for the entire course:

- **I'm having some audio and video issues, is there any recommended troubleshooting steps?** Yes, you can use the following steps provided by Udemy at:

  `https://support.udemy.com/hc/en-us/articles/229231227-Video-Audio-Issues-Troubleshooting`

  and also at:

  `https://support.udemy.com/hc/en-us/articles/360003054894-Troubleshooting-Issues-With-Udemy-s-App`

- **What IDE is the instructor using?** We are using Spyder inside Anaconda.
- **Where can I access the course materials and recommended readings?** You can visit the following link to obtain the materials

  `https://www.superdatascience.com/computer-vision`

  .
- **Which version of PyTorch will work with our algorithms and is it supported by Windows?** PyTorch does support Windows but we need to use versions ¡ 0.4.0 of PyTorch for our algorithms. You can find the installation steps, whl files and more at the following for Mac and Linux: `https://pytorch.org/previous-versions/` and for Windows at: `https://github.com/peterjc123/pytorch-scripts`.
- **Which Python version does the course use?** The course was originally built with Python 3.5. You can download all of the requirements by utilizing the YML found at `https://www.superdatascience.com/computer-vision` in the installations folder.
- **I'm running into an error installing PyTorch for Windows, what can I do to resolve it?** Please use the modified setup for

installing the YML and PyTorch on Windows that can be found at: `https://www.udemy.com/computer-vision-a-z/learn/v4/t/lecture/10609646?start=0`

- **Where can I obtain all of the required files?** You can obtain the handbook by visiting `https://www.superdatascience.com/computer-vision/` The code templates are included for each module that contain the files with the code used in the course.

Table of Contents

Summary

1 Section: 1 - Introduction **7**
1.1 Lecture 1. Welcome to the Course! . . . 7
1.2 Lecture 2. Some Additional Resources!! . . . 7

2 Section: 2 - Module 1 - Face Detection Intuition **8**
2.1 Lecture 3. Plan of attack . . . 8

3 Lecture 4. Viola-Jones Algorithm **8**
3.1 Lecture 5. Haar-like Features . . . 8
3.2 Lecture 6. Integral Image . . . 8
3.3 Lecture 7. Training Classifiers . . . 8
3.4 Lecture 8. Adaptive Boosting (Adaboost) . . . 9
3.5 Lecture 9. Cascading . . . 9
3.6 Quiz 1: Face Detection Intuition . . . 9

4 Section 3 - Module 1 - Face Detection with OpenCV **10**
4.1 Lecture 10 - Welcome to the Practical Applications . . . 10
4.2 Lecture 11 -Installations Instructions (once and for all!) . . . 10
4.3 Lecture 12 - Common Debug Tips . . . 12
4.4 Lecture 13 - Face Detection - Step 1 . . . 12
4.5 Lecture 14 - Face Detection - Step 2 . . . 12
4.6 Lecture 15 - Face Detection - Step 3 . . . 12
4.7 Lecture 16 - Face Detection - Step 4 . . . 13
4.8 Lecture 17 - Face Detection - Step 5 . . . 13
4.9 Lecture 18 - Face Detection - Step 6 . . . 13
4.10 Quiz 2: Face Detection with OpenCV . . . 13

5 Section 4 - Homework Challenge - Build a Happiness Detector **14**
5.1 Lecture 19 - Homework Challenge - Instructions . . . 14
5.2 Lecture 20 - 20. Homework Challenge - Solution (Video) . . . 14
5.3 Lecture 21 - Homework Challenge - Solution (Code files) . . . 14

6 Section 5 - Module 2 - Object Detection Intuition **15**
6.1 Lecture 22 - Plan of attack . . . 15
6.2 Lecture 23 - How SSD is different . . . 15
6.3 Lecture 24 - The Multi-Box Concept . . . 15
6.4 Lecture 25 - Predicting Object Positions . . . 16
6.5 Lecture 26 - The Scale Problem . . . 16
6.6 Quiz 3: Object Detection Intuition . . . 16

7 Section 6 - Module 2 - Object Detection with SSD 17
7.1 Lecture 27 - Object Detection - Step 1 17
7.2 Lecture 28 - Object Detection - Step 2 18
7.3 Lecture 29 - Object Detection - Step 3 18
7.4 Lecture 30 - Object Detection - Step 4 18
7.5 Lecture 31 - Object Detection - Step 5 20
7.6 Lecture 32 - Object Detection - Step 6 20
7.7 Lecture 33 - Object Detection - Step 7 20
7.8 Lecture 34 - Object Detection - Step 8 20
7.9 Lecture 35 - Object Detection - Step 9 21
7.10 Lecture 36 - Object Detection - Step 10 21
7.11 Lecture 37 - Training the SSD 21
7.12 Quiz 4 - Object Detection with SSD 21

8 Section 7 - Homework Challenge - Detect Epic Horses galloping in Monument Valley 22
8.1 Lecture 38 - Homework Challenge - Instructions 22
8.2 Lecture 39 - Homework Challenge - Solution (Video) 22
8.3 Lecture 40 - Homework Challenge - Solution (Code files) 22
8.4 Section 8 - Module 3 - Generative Adversarial Networks (GANs) Intuition 23
8.5 Lecture 41 - Plan of Attack 23
8.6 Lecture 42 - The Idea Behind GANs 23
8.7 Lecture 43 - How Do GANs Work? (Step 1) 24
8.8 Lecture 44 - How Do GANs Work? (Step 2) 24
8.9 Lecture 45 - How Do GANs Work? (Step 3) 24
8.10 Lecture 46 - Applications of GANs 25
8.11 Quiz 5: Generative Adversarial Networks (GANs) Intuition . . . 25

9 Section 9 - Module 3 - Image Creation with GANs 26
9.1 Lecture 47 - GANs - Step 1 26
9.2 Lecture 48 - GANs - Step 1 26
9.3 Lecture 49 - GANs - Step 2 26
9.4 Lecture 50 - GANs - Step 3 27
9.5 Lecture 51 - GANs - Step 4 27
9.6 Lecture 52 - GANs - Step 5 27
9.7 Lecture 53 - GANs - Step 6 27
9.8 Lecture 54 - GANs - Step 7 27
9.9 Lecture 55 - GANs - Step 8 27
9.10 Lecture 56 - GANs - Step 9 27
9.11 Lecture 57. GANs - Step 10 27
9.12 Lecture 58. GANs - Step 11 27
9.13 Quiz 6: Image Creation with GANs 28
9.14 Lecture 60 - 60. Special Thanks to Alexis Jacq 28

10 Section 10 - Annex 1: Artificial Neural Networks **29**
10.1 Lecture 61 - What is Deep Learning? 29
10.2 Lecture 62 - Plan of Attack 29
10.3 Lecture 63 - 63. The Neuron 29
10.4 Lecture 64 - 64. The Activation Function 29
10.5 Lecture 65 - How do Neural Networks work? 29
10.6 Lecture 66 - How do Neural Networks learn? 29
10.7 Lecture 67 - Gradient Descent 29
10.8 Lecture 68 - Stochastic Gradient Descent 29
10.9 Lecture 69 - Backpropagation 29

11 Section 11 - Annex 2: Convolutional Neural Networks **30**
11.1 Lecture 70 - Plan of Attack 30
11.2 Lecture 71 - What are convolutional neural networks? 30
11.3 Lecture 72 - Step 1 - Convolution Operation 30
11.4 Lecture 73 - Step 1(b) - ReLU Layer 30
11.5 Lecture 74 - Step 2 - Pooling 30
11.6 Lecture 75 - Step 3 - Flattening 30
11.7 Lecture 76 - Step 4 - Full Connection 30
11.8 Lecture 77 - Summary 30
11.9 Lecture 78 - Softmax & Cross-Entropy 30

12 Conclusion **31**

1 Section: 1 - Introduction

1.1 Lecture 1. Welcome to the Course!

- **Where can I find the actual code materials for this course??** You can find all of the materials for this course at the following link: `https://www.superdatascience.com/computer-vision/`.If you have any questions please feel free to post them in the Q&A.

1.2 Lecture 2. Some Additional Resources!!

- **Congrats on enrolling in the Deep Learning & Computer Vision A-Z course! In order to ease in to this amazing field, we've selected a great episode you can listen to on your commute, at breakfast or wherever to get you fired up to continue in this fantastic journey. Click here to get started:**

`https://www.superdatascience.com/podcast-unstoppable-data-science-reckless-commitment-artificial-intelligence`

2 Section: 2 - Module 1 - Face Detection Intuition

2.1 Lecture 3. Plan of attack

3 Lecture 4. Viola-Jones Algorithm

3.1 Lecture 5. Haar-like Features

- **Will this course have different face recognition techniques with deep learning?** In this course we cover face detection in the first module, in the second using SSD for detection and GAN's in the third module. You can visit `https://www.superdatascience.com/computer-vision/` for the recommended readings and course materials as well but you can also use the first module and extend it or the SSD for further detection related algorithms.

3.2 Lecture 6. Integral Image

- **Is isolating the pixels that make up the feature, an operation that is computationally expensive that uses multiplications for box filtering, which means that calculating summations is much less expensive than calculating multiplications?** Correct, for working with image processing we rely on features specific to certain regions of the image and need the properties. Take a look here because the example does a nice job of clarifying:

  Find out more here

  In that example we can calculate without integral image and it costs 900 operations. If we use integral image it only costs 56 operations. To quote that explanation in the article "For just hundred operations over a 5 x 5 matrix, using an integral image uses about 50 % less computations, imagine the amount of difference it makes for large images and more such operations." You can also see it used here in this for some additional information:

  Find out more here

3.3 Lecture 7. Training Classifiers

- **How can I build a new classifier?** Please see the following that will help provide you with some information on how to approach putting that together:

  `https://stackoverflow.com/questions/2000816/how-to-create-haar-cascade-xml-for-using-with-opencv`.

You can also find the other haar examples at: `https://github.com/opencv/opencv/tree/master/data/haarcascades`.

3.4 Lecture 8. Adaptive Boosting (Adaboost)

- **How are the number of features 18000?**f we look at our 24x24 picture we can see that the features are scalable and can have variations. If you can take into consideration all of the possibilities of widths and heights of the feature and all of the possible variations in the different positions add up to 180,000 possible explorations.
- **Can we use CNN as it might give the better result instead of haar-like feature detection?** Yes, you can absolutely use a CNN. That is what you will learn in Module 2 which contains an advanced CNN-like neural network, called the SSD - Single Shot Multi Box Detection. But we wanted to include OpenCV for their simplicity of understanding and use.

3.5 Lecture 9. Cascading

- **Why is it Non-Deterministic Search?**With the non-deterministic search we are modeling a scenario that is basically out of control of the agent (the probabilities) that can't be predicted.
- **How did we arrive at calculating the probability of each state?** As an overview for the values in the states is that we are using the MDP to calculate the value of each state but in this scenario we know the optimal route or can calculate a path for the agent to take working back from the final reward of 1. Normally when the agent would go through this scenario it would run multiple iterations and learn while establishing the values (reinforcement learning).

3.6 Quiz 1: Face Detection Intuition

4 Section 3 - Module 1 - Face Detection with OpenCV

4.1 Lecture 10 - Welcome to the Practical Applications

- **Why is the virtual environment required? Can't we do all these without a virtual environment?** You can set it up and install all of the packages on their own without the virtual environment but it was used to make it an easier setup process with one command to install all of the requirements to run the algorithms by setting up the virtual environment. It's also a great method to use different environments in Anaconda to be able to run specific versions of Python and other libraries.

- **The Computer Vision folder that I downloaded is empty, where are the files?** If the folder is empty it might be the download for the folder structure that is used to organize the modules and templates. The template folder is needed for the files. You can download the code templates (the files for each section) that you can find if you scroll down the page `https://www.superdatascience.com/computer-vision/`,for example it will say Code Templates: Face Detection. The code templates have the files that you can then add to the folder structure. If you have any trouble accessing the files from the templates please just let me know!

4.2 Lecture 11 -Installations Instructions (once and for all!)

- **I have the following installation error on Windows, ResolvePackageNotFound: - pytorch==0.1.12=py35**$_0$.1.12$cu80-WhatcanIdotoresolveit?$

```
Can you try the following setup to resolve it:
(if you dont have Nvidia drivers please us the CPU version)

# If your main Python version is not 3.5 or 3.6
conda create -n test python=3.6 numpy pyyaml mkl

# for CPU only packages
conda install -c peterjc123 pytorch-cpu

# for Windows 10 and Windows Server 2016, CUDA 8
conda install -c peterjc123 pytorch

# for Windows 10 and Windows Server 2016, CUDA 9
conda install -c peterjc123 pytorch cuda90

# for Windows 7/8/8.1 and Windows Server 2008/2012, CUDA 8
```

```
conda install -c peterjc123 pytorch_legacy
You can use the above commands to change depending on the OS for #6

If the virtual_platform environment was created please remove it with:
conda env remove -n virtual_platform.
You can go into the Anaconda Navigator and click on the environments
tab to see if it's listed.

1. Open the YML file in an editor

2. Delete: line 92 (i.e. the line: - pytorch=0.1.12=py35_0.1.12cu80)
and line 100 (- torch==0.1.12) and line 101 (- torchvision==0.1.9)

 3. Save

4. Execute: conda env create -f virtual_platform_windows.yml

5. Activate virtual environment: source activate virtual_platform

6. Install Pytorch seperately with:
   conda install -c peterjc123 pytorch cuda80

7. pip install --no-deps torchvision
```

- **I'm receiving the following error for Open CV: ModuleNotFoundError: No module named 'cv2'. What can I do?** Please use the following steps to install OpenCV:

```
To install OpenCV for the error please try running:

(if you are using the virtual_platform first activate it and then run)

Mac/Linux:

source activate virtual_platform

Windows

activate virtual_platform

and then:

pip install opencv-python
```

4.3 Lecture 12 - Common Debug Tips

- **Please check this document often as it is continuously updated to include the most recent debug tips and modifications.**

4.4 Lecture 13 - Face Detection - Step 1

- **Why are we using OpenCV?**I think in the simplest terms is that OpenCV is the most prominent and widely used computer vision related library. In addition with it's support of Python and C++ along with it's features it allows for algorithms such as face detection to be implemented efficiently. OpenCV still releases updates and stable builds (they released 3.4.1 in February)

- **Is there any other information on creating the haar cascade file?**I always recommend checking the official documentation as a starting point: `http://docs.opencv.org/2.4/doc/user_guide/ug_traincascade.html` but in addition you can take a look here as well for some info:

  `https://stackoverflow.com/questions/2000816/how-to-create-haar-cascade-xml-for-using-with-opencv`

4.5 Lecture 14 - Face Detection - Step 2

- **What is the mission of the minneighbours parameter??**For min neighbors, in order for a zone of pixels to be accepted we need a certain number of neighbor zones accepted (the min neighbors we need 5 neighbor zones). The numbers we are using was learned from experimenting with the values.

- **Can you please explain the use of the argument '5' which inputs the number of neighbour pixels to be included?**n order for a zone of pixels to be accepted you have to have a certain number of neighbor zones that have to be accepted. We have to use the minimum number so we are using 5 in this case (5 neighbor zones to be accepted) and through experimentation it was learned that we get great results with 1.3, 5. This might help give you some additional information:

  `https://docs.opencv.org/3.0-beta/modules/imgproc/doc/feature_detection.html`

4.6 Lecture 15 - Face Detection - Step 3

- **How is the y-coordinates of the lower right corner y+h?** It's due to the coordinate-system that is a numpy array, like a matrix. In a matrix you start from the upper left corner, then the positive x-coordinates move you to the right and the positive y-coordinates move you down. So y+h is at the bottom, not the top.

4.7 Lecture 16 - Face Detection - Step 4

4.8 Lecture 17 - Face Detection - Step 5

- **How can we get other readily available XML files?** It's possible to create custom ones but for readily available XML files they can be found here: `https://github.com/opencv/opencv/tree/master/data/haarcascades`.

- **How can we create our own XML files?**Please see the following for an example on creating custom XML files: `http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html`.

4.9 Lecture 18 - Face Detection - Step 6

- **Can you explain the function of 0xFF condition?**Please take a look at the following since it will help clarify the use of 0xff: `https://stackoverflow.com/questions/35372700/whats-0xff-for-in-cv2-waitkey1/39201163`.

- **I'm receiving the following error, modules/objdetect/src/cascadedetect.cpp:1639: error: (-215) !empty() in function detectMultiScale. How can I resolve it?**Please try the solutions mentioned in the following discussion: `https://www.udemy.com/computer-vision-a-z/learn/v4/questions/4024852`.

4.10 Quiz 2: Face Detection with OpenCV

5 Section 4 - Homework Challenge - Build a Happiness Detector

5.1 Lecture 19 - Homework Challenge - Instructions

5.2 Lecture 20 - 20. Homework Challenge - Solution (Video)

- **What are neighbors in OpenCV, or in general?** Jordan — Teaching Assistant · 10 months ago Hi Prince,

  Usually when we reference neighbors (even in AI, ML, Data Science etc) in general when you reference the term it is looking to find the closest data points or values to a specific point (neighbors). For example, take a look at the following from OpenCV and it should help provide you some additional information for further clarification:

  `https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_ml/py_knn/py_knn_understanding/py_knn_understanding.html` and also `https://github.com/opencv/opencv/search?utf8=%E2%9C%93&q=neighbors&type=`.

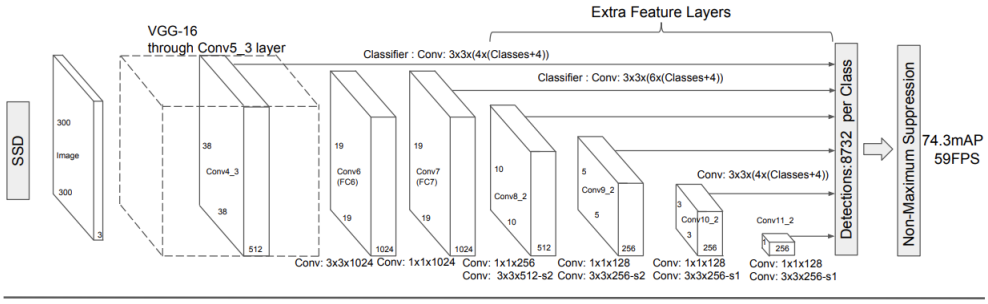
5.3 Lecture 21 - Homework Challenge - Solution (Code files)

6 Section 5 - Module 2 - Object Detection Intuition

6.1 Lecture 22 - Plan of attack

6.2 Lecture 23 - How SSD is different

- **please can you explain me the meaning of the different notation in the SSD architecture, for example: 1- conv4_3, 2- conv6 (FC6), 3- classifier: conv: 3X3(4X(classes + 4))** Those examples are referencing specific parts of the architecture, extra feature layers, convolutional layers, reducing the object size in each convolution etc - for example:



This can also be used to examine the architecture as well: `https://www.cs.unc.edu/~wliu/papers/ssd.pdf`

- **Is Yolo or SSD better?**Currently SSD is a better choice and has a lot of research involving and supporting it but YOLO2 and YOLO3 recently released are growing quite quickly. It's something to pay attention too and just goes to show how awesome the world of deep learning and computer vision is since it's such an active area of research.

6.3 Lecture 24 - The Multi-Box Concept

- **SSD go through the images only once. But @7:58 its said that the error is back propagated.Doesn't that mean that image is accessed more that one time?** We are seeing the error being back-propagated through the network so that the weights are updated. This helps with updating so that the box trains over the iterations for the prediction.

6.4 Lecture 25 - Predicting Object Positions

6.5 Lecture 26 - The Scale Problem

- **Hi, just wanted to check how is the image represented as 300 X 300 X 3? I can understand 300px might be the height/width but how does 3 come into picture?** The following image as the 300x300x3 is used for the input size, the dimension and you can see how it transforms through each layer if you look at each section of the image. For example:

6.6 Quiz 3: Object Detection Intuition

7 Section 6 - Module 2 - Object Detection with SSD

7.1 Lecture 27 - Object Detection - Step 1

- **I'm receiving the following error: ValueError: not enough values to unpack (expected 2, got 0). What can I do to resolve it?** Please open the terminal or prompt and if you are using a virtualplease activate it with:

```
Mac/Linux:
Source Activte my\_env\_name

Windows
Activate my\_env\_name

Then please run pip list or conda list to check the version of PyTorch.
If it's 0.4.0 (which has introduced this bug)
to resolve it you can downgrade to any version < 0.4.0.

For Mac/Linux you can find the versions here:
https://pytorch.org/get-started/previous-versions/

For Windows:

# If your main Python version is not 3.5 or 3.6
conda create -n test python=3.6 numpy pyyaml mkl

# for CPU only packages
conda install -c peterjc123 pytorch-cpu

# for Windows 10 and Windows Server 2016, CUDA 8
conda install -c peterjc123 pytorch

# for Windows 10 and Windows Server 2016, CUDA 9
conda install -c peterjc123 pytorch cuda90

# for Windows 7/8/8.1 and Windows Server 2008/2012, CUDA 8
conda install -c peterjc123 pytorch_legacy
```

- **I'm receiving the following error: ImportError: DLL load failed: The specified procedure could not be found. What can I do to resolve it?** For the DLL error, it's normally caused by the version of PyTorch. If you don't have Nvidia drivers please re-install PyTorch with

the CPU version. If it persists, please follow the steps listed here: `https://pytorch.org/docs/stable/notes/windows.html#import-error`.

7.2 Lecture 28 - Object Detection - Step 2

- **Why are we using PyTorch over Tensorflow?** Using PyTorch helps considerably speed up the training computations, and the predictions as well. Comparing it to Keras, it is way faster, more efficient and more powerful. It can also come down to architecture choice and PyTorch can be easier to work with at times to implement specific modules such as the SSD.

- **I'm receiving an error stating the OpenCV module can't be found. What steps can I use to resolve it?**

  If you are using the virtual platform please activate it first with:

  Mac/Linux

  ```
  source activate virtual_platform
  ```

  Windows

  ```
  activate virtual_platform
  ```

  Then please run:

  ```
  pip install opencv-python
  ```

7.3 Lecture 29 - Object Detection - Step 3

- **What does transforming the image to be compatible with the Neural Network mean?**It's referring to convert the size of the images, put them in the right dimensions, the right scales of colour values, according to the convention under which the neural network was trained.

7.4 Lecture 30 - Object Detection - Step 4

- **I received an error stating no module named 'data'. How can I resolve it?** Please double check the working environment includes all of the files such as discussed here: `https://www.udemy.com/computer-vision-a-z/learn/v4/questions/4172716`.

- **What do we use the unsqueeze function for?**We use the unsqueeze function to create the fake dimension of the batch (our third transformation) since the neural network can't accept a single input image because it only accepts batches of inputs. When you see unsqueeze in a neural

network it's before fitting the nn with input. For our argument we need to use the index of the dimension of the batch (0). You can see some further information at: `http://pytorch.org/docs/master/tensors.html?highlight=unsqueeze#torch.Tensor.unsqueeze`.

- **Why are we changing numpy array into tensor variable?** Converting it to a torch tensor allows us to work with a more useful (due to it's matrix setup) with a neural network compared to a numpy array and allows us to use the methods we further implement. It can always come down to development technique as well, but when working with a neural net, pytorch tensors due have a big benefit, for example for some additional context: `http://pytorch.org/docs/master/tensors.html`.

torch.Tensor

A `torch.Tensor` is a multi-dimensional matrix containing elements of a single data type.

Torch defines eight CPU tensor types and eight GPU tensor types:

Data type	dtype	CPU tensor	GPU tensor
32-bit floating point	`torch.float32` or `torch.float`	`torch.FloatTensor`	`torch.cuda.Flo`
64-bit floating point	`torch.float64` or `torch.double`	`torch.DoubleTensor`	`torch.cuda.Dou`
16-bit floating point	`torch.float16` or `torch.half`	`torch.HalfTensor`	`torch.cuda.Hal`
8-bit integer (unsigned)	`torch.uint8`	`torch.ByteTensor`	`torch.cuda.Byt`
8-bit integer (signed)	`torch.int8`	`torch.CharTensor`	`torch.cuda.Cha`
16-bit integer (signed)	`torch.int16` or `torch.short`	`torch.ShortTensor`	`torch.cuda.Sho`
32-bit integer (signed)	`torch.int32` or `torch.int`	`torch.IntTensor`	`torch.cuda.Int`
64-bit integer (signed)	`torch.int64` or `torch.long`	`torch.LongTensor`	`torch.cuda.Lon`

`torch.Tensor` is an alias for the default tensor type (`torch.FloatTensor`).

- **Is there any other information on realtime detection?** Please take a look at the following discussion for some info on real time detection: `https://www.udemy.com/computer-vision-a-z/learn/v4/questions/3513956` and also at `https://www.udemy.com/computer-vision-a-z/learn/v4/questions/342544`.

7.5 Lecture 31 - Object Detection - Step 5

- **What is the intutition behind normalization?**It's due to the need to specify the dimensions of the tensor (and the locations of the rectangle) to normalize the scale of objects/values of the detected objects. Our new tensor object that has 4 dimensions w/h/w/h is created due to the position of the detected objects within the image has to be normalized between 0 - 1. This tensor (scale) normalizes the objects detected in the image. W/H is for the upper left hand corner of the rectangle detector and the second W/H for the lower right for the same rectangle detector.

  scale = Torch.tensor([Width, Height, Width, Height])

  As for using the boxes, have a look at this additional reference when you get the chance :

  `https://www.learnopencv.com/how-to-select-a-bounding-box-roi-in-opencv-cpp-python/`.

7.6 Lecture 32 - Object Detection - Step 6

- **At code line 20, which is a comment, could you share more clarification as to understand what is "the batch"?**The tensor contains 4 elements (the first being the batch) and it was created with unsqueeze and is our outputs. For some additional information and examples see the following:

  `https://www.quora.com/What-is-meant-by-Batch-in-machine-learning`
  and
  `https://stackoverflow.com/questions/41175401/what-is-a-batch-in-tensorflow`.

7.7 Lecture 33 - Object Detection - Step 7

- **I didn't fully understand the line: map_location = lambda storage, loc: storage. Can you please elaborate further on it?**. The map_location = lambda storage, loc: storage is used to remap the storage of all tensors into the CPU. If you would like some additional information please see the following: `https://discuss.pytorch.org/t/on-a-cpu-device-how-to-load-checkpoint-saved-on-gpu-device/349`.

7.8 Lecture 34 - Object Detection - Step 8

-

7.9 Lecture 35 - Object Detection - Step 9

- **Please see the following for information on using CUDA for the SSD:** `https://www.udemy.com/computer-vision-a-z/learn/v4/t/lecture/8127652?start=0`.

7.10 Lecture 36 - Object Detection - Step 10

- **I received the error: RuntimeError: sum() missing 1 required positional arguments: "dim". How can I resolve it?**Please try changing the following:

  Change
  ```
  num_pos = pos.sum(keepdim=True)
  ```
  To
  ```
  num_pos = pos.sum(dim=1, keepdim=True)
  ```

7.11 Lecture 37 - Training the SSD

7.12 Quiz 4 - Object Detection with SSD

8 Section 7 - Homework Challenge - Detect Epic Horses galloping in Monument Valley

8.1 Lecture 38 - Homework Challenge - Instructions

8.2 Lecture 39 - Homework Challenge - Solution (Video)

8.3 Lecture 40 - Homework Challenge - Solution (Code files)

8.4 Section 8 - Module 3 - Generative Adversarial Networks (GANs) Intuition

8.5 Lecture 41 - Plan of Attack

8.6 Lecture 42 - The Idea Behind GANs

- **Is there a main difference between SSDs and GANs?**

  SSD –

  Key idea here is single network (for speed) and no need for region proposals instead it uses different bounding boxes and then adjust the bounding box as part of prediction. Different bounding box predictions is achieved by each of the last few layers of the network responsible for predictions for progressively smaller bounding box and final prediction is union of all these predictions.

  SSD, discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. The fundamental improvement in speed comes from eliminating bounding box proposals and the subsequent pixel or feature resampling stage.

  Find out more here

  GANs –

  Generative adversarial networks (GANs) are deep neural net architectures comprised of two nets, pitting one against the other (thus the adversarial).

  GANs were introduced in a paper by Ian Goodfellow and other researchers at the University of Montreal, including Yoshua Bengio, in 2014. Referring to GANs, Facebooks AI research director Yann LeCun called adversarial training the most interesting idea in the last 10 years in ML.

  GANs potential is huge, because they can learn to mimic any distribution of data.

```
That is, GANs can be taught to create worlds
eerily similar to our own in any domain: images,
music, speech, prose. They are robot artists in
a sense, and their output is impressive, poignant even.

https://deeplearning4j.org/generative-adversarial-network
```

```
If you want to look at some similar algorithms to SSD,
you could compare it to the use of YOLO, YOLO2/3 or an
RNN, but each has their own advantages and disadvantages
related to the use case.
```

8.7 Lecture 43 - How Do GANs Work? (Step 1)

- **Why does the discriminator need to learn on the new generated images?** This reference does a great job of explaining the information to help clarify that question and it has a Spongebob reference which is always a plus:

  Click Here

  That being said a key take away from that article is that: "Through training, the discriminator network learns a function to tell the difference between the real and generated data. The first features the discriminator learns to look for may be relatively obvious aspects of the data which easily separate the real from the fake"

8.8 Lecture 44 - How Do GANs Work? (Step 2)

8.9 Lecture 45 - How Do GANs Work? (Step 3)

- **Are the same real dog images shown to the Discriminator on each 'step/round' for the Discriminator to compare with the Generated ones or always different from a huge library of real dog images ?** ""A Generative Adversarial Network works through the interplay between two semi-separate networks: a generator and a discriminator. The goal of the discriminator is to tell the difference between the data generated by the generator and the real-world data we are trying to model. We can think of the discriminator as being like the bouncer at a club. I don't have just any bouncer in mind though. Discriminator networks are like the bouncer outside the Salty Spitoon, as seen in the Spongebob Squarepants episode:`http://spongebob.wikia.com/wiki/No_Weenies_Allowed`.

In the world of Spongebob Squarepants, one must be tough in order to get into the Salty Spitoon. The job of the bouncer (discriminator) is to tell the difference between the real tough guys (the real data), and the weenies (the generated data).

Then we have Spongebob Squarepants (the generator). He certainly isn't a real tough guy. He is an imitator, and needs to learn to look like a tough guy in order to get into to see his friend."...

If you get the chance, take a look for additional information here:
Click Here

8.10 Lecture 46 - Applications of GANs

8.11 Quiz 5: Generative Adversarial Networks (GANs) Intuition

9 Section 9 - Module 3 - Image Creation with GANs

9.1 Lecture 47 - GANs - Step 1

- **Why did we define the weight initialization function? Also, what is the default value of initialization function in PyTorch?** We use the weight initialization so that it will take a nn to initialize the weights of the nn (applied to both the nn of the generator and the nn of the discriminator) for the adversarial networks.As for the default initialization the following will provide some further context for PyTorch:

  https://discuss.pytorch.org/t/whats-the-default-initialization-methods-for-layers/3157 and also

  https://discuss.pytorch.org/t/how-are-layer-weights-and-biases-initialized-by-default/13073.

9.2 Lecture 48 - GANs - Step 1

- **What is the meaning of .bias.data.fill_(0)?** This is used to initialize the bias within the linear layer of the neural network. For example please see these two additional examples as they will help provide some further context:

  https://discuss.pytorch.org/t/how-are-biases-initialized-in-pytorch-in-the-linear-layer/5049/5 and also https://github.com/pytorch/pytorch/blob/master/torch/nn/modules/linear.py#L44-L48.

- **Could you please share papers based on which this architecture is adopted and implemented?** The architecture created in Module 3 is inspired from this paper: https://arxiv.org/pdf/1511.06434.pdf.

9.3 Lecture 49 - GANs - Step 2

- **How we can pass input to a Class G? Is that really possible to pass arguments like that?**Yes it is because the forward function is connected to the __init__ function via "self". In that case you don't have to specify the name of the method. But in general case you do have to specify the name of the method following the object and a dot, like this:

```
output = netG.forward(input)
```

9.4 Lecture 50 - GANs - Step 3

9.5 Lecture 51 - GANs - Step 4

- **Is the article of experiment of the figures available?** It's from the following paper: `https://arxiv.org/pdf/1511.06434.pdf`.

9.6 Lecture 52 - GANs - Step 5

9.7 Lecture 53 - GANs - Step 6

9.8 Lecture 54 - GANs - Step 7

9.9 Lecture 55 - GANs - Step 8

9.10 Lecture 56 - GANs - Step 9

- **Why don't we use netD.foward to get the output?** It is being forward propagated, we are feeding the nn with the input (a mini batch of real images) and through the main module that is forward propagating the real images inside the mini batch -

```
output = netD(input) # We forward propagate this real
image into the neural network of the discriminator to get the
prediction (a value between 0 and 1).

output = netD(fake.detach()) # We forward propagate the fake generated
images into the neural network of the discriminator to get
the prediction (a value between 0 and 1).
```

- **What do you mean a tensor also has attached gradients?** We need to (in the discriminator) take the nn to use zero_grad() that automatically initializes the gradients with respect to the weights.

  The following might help with tensors and gradients: `https://pytorch.org/tutorials/beginner/examples_autograd/two_layer_net_autograd.html`.

9.11 Lecture 57. GANs - Step 10

9.12 Lecture 58. GANs - Step 11

- **Can I run the GAN with CUDA?** Yes, you can run it with CUDA. Please see the following for some further information: `https://www.udemy.com/computer-vision-a-z/learn/v4/questions/3096984`

9.13 Quiz 6: Image Creation with GANs

9.14 Lecture 60 - 60. Special Thanks to Alexis Jacq

10 Section 10 - Annex 1: Artificial Neural Networks

10.1 Lecture 61 - What is Deep Learning?

10.2 Lecture 62 - Plan of Attack

10.3 Lecture 63 - 63. The Neuron

10.4 Lecture 64 - 64. The Activation Function

- **Could you provide any other information on using sigmoid functions for neural nets?** Absolutely, please see the following useful discussion:

  `https://stats.stackexchange.com/questions/162988/why-sigmoid-function-instead-of-anything-else`

10.5 Lecture 65 - How do Neural Networks work?

10.6 Lecture 66 - How do Neural Networks learn?

10.7 Lecture 67 - Gradient Descent

10.8 Lecture 68 - Stochastic Gradient Descent

10.9 Lecture 69 - Backpropagation

- **Can you provide any further information for the learning rate?** Learning rate does play an important role:"Learning rate is a hyperparameter that controls how much we are adjusting the weights of our network with respect the loss gradient. The lower the value, the slower we travel along the downward slope. While this might be a good idea (using a low learning rate) in terms of making sure that we do not miss any local minima, it could also mean that we'll be taking a long time to converge—especially if we get stuck on a plateau region"

  https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10

  In addition, see the following on learning rate in Neural Nets (Back Propagation) by Andrew Ng:`http://web.stanford.edu/class/cs294a/sparseAutoencoder_2011new.pdf`

11 Section 11 - Annex 2: Convolutional Neural Networks

11.1 Lecture 70 - Plan of Attack

11.2 Lecture 71 - What are convolutional neural networks?

11.3 Lecture 72 - Step 1 - Convolution Operation

11.4 Lecture 73 - Step 1(b) - ReLU Layer

11.5 Lecture 74 - Step 2 - Pooling

11.6 Lecture 75 - Step 3 - Flattening

11.7 Lecture 76 - Step 4 - Full Connection

11.8 Lecture 77 - Summary

11.9 Lecture 78 - Softmax & Cross-Entropy

12 Conclusion

Please check back often as we will continuously update the FAQ document, and if you have any questions please feel free to post in the course Q&A.