**Windows Installation/setup guide for Activity Dataset Model**

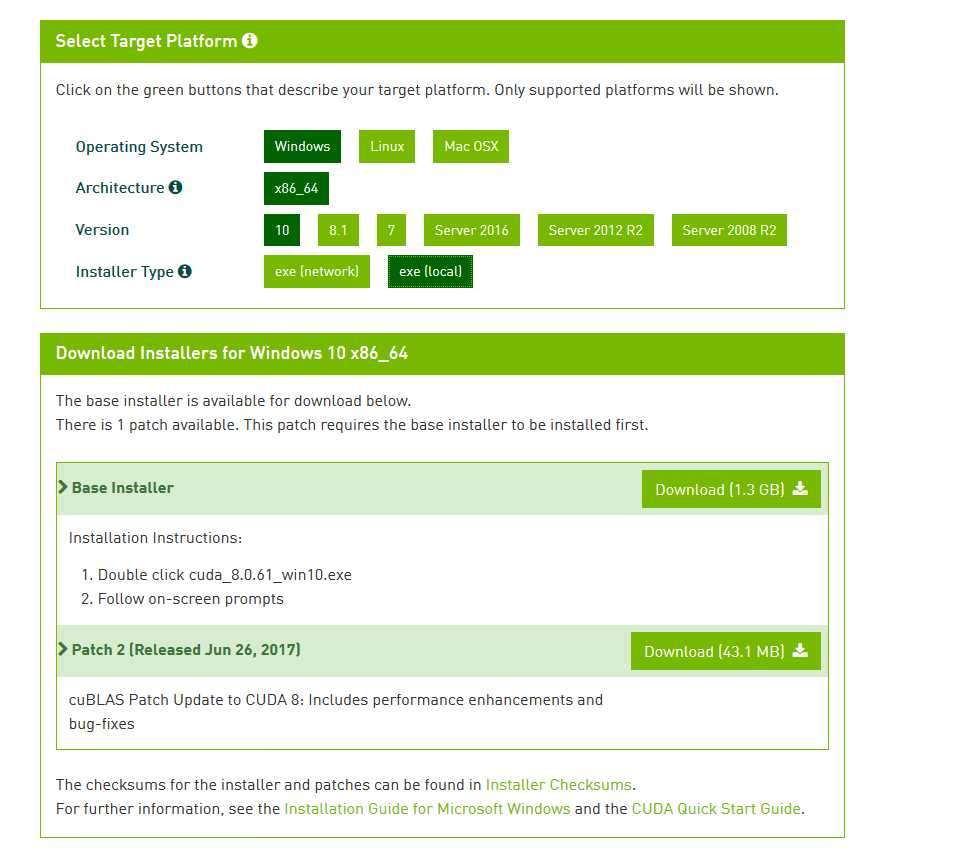
**By Andrew Grant - August 2017**

**1. Forward**

**This guide was written for installation on Windows OS, and is tested on a Windows 10 64bit installation with a Nvidia GPU. As far as I can tell, a GPU is needed to run this model. As Nvidia’s CUDA and CUDNN is needed, it is possible that a Nvidia GPU is necessary. However, I am unsure about that as I have not tested or looked into using this model on a machine with and AMD or other GPU, but it is worth keeping in mind.**

**2. Pre-Requirements – (Install in this order)**

* **Python 2.7** - <https://www.python.org/downloads/windows/>
* **Visual Studio 2015 installation with C language packages installed.** (Students can obtain a copy from here <http://onthehub.com/> - sign in with your school and search for visual studio enterprise 2015 with update 3 - you want the web installer)
* **Nvida CUDA v8.0** (Get the patch as well) <https://developer.nvidia.com/cuda-downloads> :



* **Nvidia CUDNN v5.1** (Requires registering a free account) - <https://developer.nvidia.com/cudnn> - **!!!IMPORTANT THAT YOU GET v5.1!!!**
* **OpenCV 3.3.0** - <http://opencv.org/releases.html>
* **Miniconda for Python 2.7** <https://conda.io/miniconda.html>

**2.1 Pre-Requirements Installation Notes**

For the most part installation of these requirements should be fairly straight forward as most of them come with an .exe.

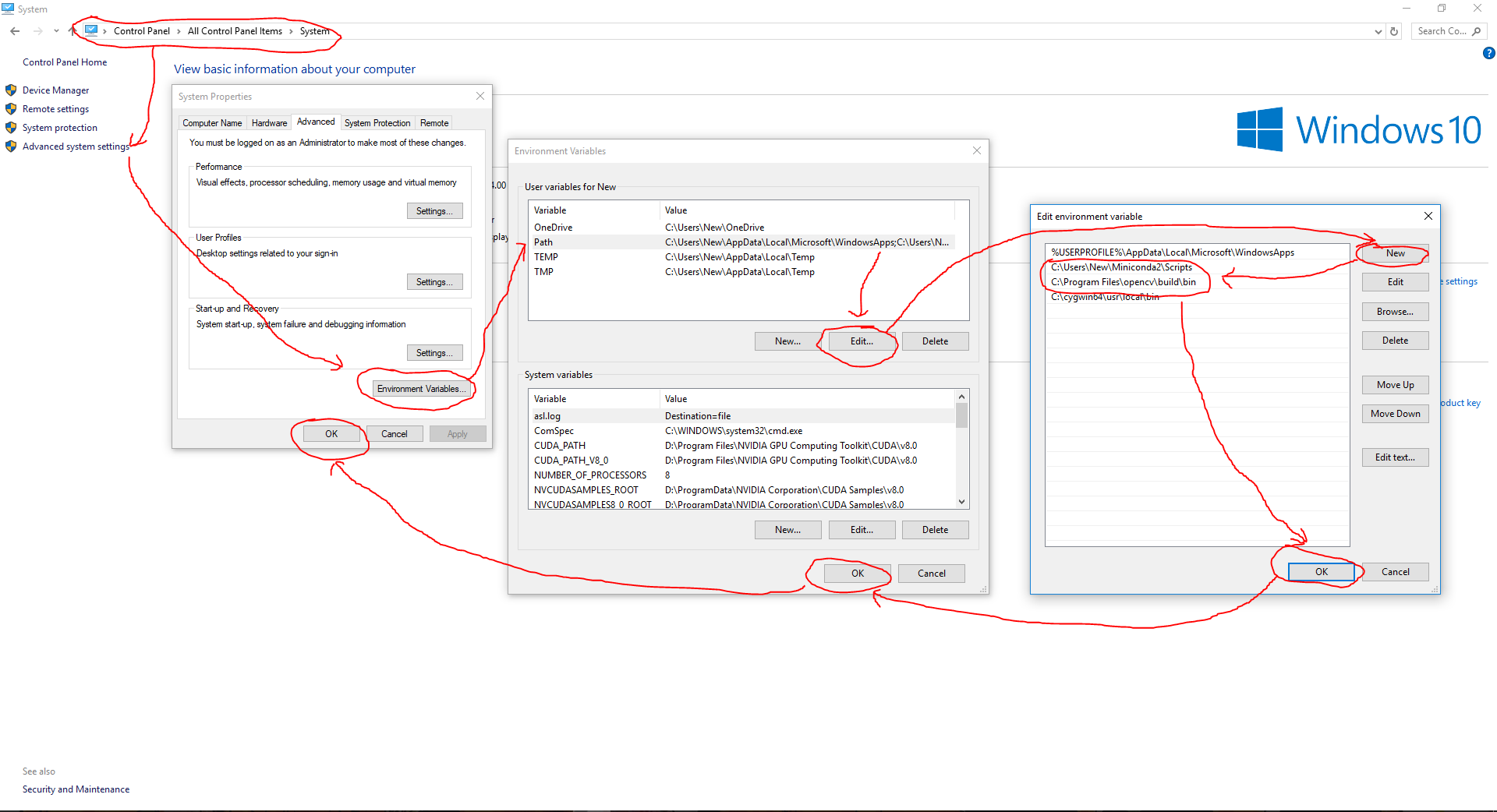
When installing Visual Studio make sure you install the C language packages. You should end up with a similar to this “*C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\bin”*

The CUDNN download should be a .zip archive with a “*cuda”* folder. Unpack the archive and open the *“cuda”* folder and there should be 3 folders “*lib”, “include”* and “*bin”* like so:



Cut and paste these folders into your Cuda v8.0 installation directory which by default should look something like this: *“C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0”.* There should already be “*lib”, “include”* and *“bin”* folders from when CUDA was installed so just merge the folders when you paste. There shouldn’t be any clashes with existing files but let CUDNN files overwrite the pre-existing CUDA files if there are.

After installing OpenCV and Miniconda you will probably need to add the *“C:\Program Files\opencv\build\bin”* andthe ***“****C:\Users\****YOUR\_USER\_NAME****\Miniconda2\Scripts”* directory and the to your *PATH environment user variable. If you don’t know how to do the image bellow (you may have to zoom in):*



**Keep in mind that you may have installed OpenCV or Miniconda to a different directory to me so make sure the paths you’re entering are correct for your setup.**

**3. Installing Theano and its dependencies on a conda virtual environment**

**NOTE: Something to keep in mind is that at the time of writing this guide the Theano version is 9.0. The next version of Theano, v10.0, will change its API calls to CUDA, so in order to use Theano with this model (which requires Cuda) it is important that the version of Theano that you install on your virtual environment is version 9.0. But as of the writing of this guide (August 2017) v9.0 is the default so you won’t have to worry about it and can just follow the instructions bellow and everything will (hopefully…) work out.**

First open up a windows command prompt terminal and enter the command:

**conda create --name capstone python=2.7**

This will create a conda virtual environment named *“capstone”* which should be installed to your “*envs”* directory here: ***“****C:\Users\****YOUR\_USER\_NAME****\Miniconda2\envs”*. We have chosen to name the environment *“capstone”* but you can name the environment anything you want, but it should be simple and easy to remember off the top of your head.

**NOTE: If the conda command fails either you incorrectly installed Miniconda, forgot to add the *“C:\Users\YOUR\_USER\_NAME\Miniconda2\Scripts”* directory to your *PATH environment variable,* or you have a different installation directory and you need to change your *PATH* edit to reflect your installation directoy. For more conda help see their guides here:** [**https://conda.io/docs/user-guide/install/windows.html**](https://conda.io/docs/user-guide/install/windows.html)

Next you will need to activate your new environment so we can install Theano and its dependencies onto it. Type in to the terminal:

**activate capstone**

You will know if your virtual environment has been activated because the name of the environment should now appear at the start of the line. For example:



Obviously if you named your virtual environment something different than *“capstone”,* let’s say *“project”* then you would instead type:**activate project** into the terminal.

**IMPORTANT: From now on, whenever you open the Command Prompt Terminal to work on the model, you will need to type:**

**activate capstone**

**Otherwise you will not be working within the virtual environment**

Now, with your environment activated we will install the Theano dependencies. Type (OR simply cut and paste):

**conda install numpy scipy mkl nose sphinx pydot-ng mkl-service libpython m2w64-toolchain nose-parameterized**

You will have to type **y** to confirm the installation of these packages.

When those packages have finished installing type:

**conda install theano pygpu**

Theano should now be installed correctly in your conda virtual environment.

Next, navigate to your user directory, presumably “*C:\Users\****YOUR\_USER\_NAME***”

**Create a .txt file named “*.theanorc*”** 

Open it in notepad or a text editor of your choice and paste in the following:

**[global]**

**floatX = float32**

**device = gpu**

**mode=FAST\_RUN**

**optimizer\_including = cudnn**

**[cuda]**

**root = C:\Program Files\NVIDIA Corporation\Installer2\CUDAToolkit\_8.0.{CA5A99DE-972D-4125-BEA7-02ACB45F87AC}**

**[nvcc]**

**flags = -LD:\Miniconda2\libs**

**compiler\_bindir = C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\bin**

**fastmath = True**

**IMPORTANT: Before you save and close the file, it is important that you ensure the highlighted paths are the same on your system. If not change them to match your installation directories. If you don’t have those directories anywhere your system that you didn’t install CUDA or Visual Studio 2015 correctly.**

**4. Setting up the Activity Dataset Model**

To save time I have compiled a .rar archive of model that is setup for our purposes already which can be downloaded from my dropbox here: <https://www.dropbox.com/sh/7y7n7023rd0hv0m/AAB4Va_aInDZXPbtOe1l7Hhfa?dl=0>

Unpack the archive to an easy to access directory. I choose to unpack the model into my user directory as it is my command prompt terminal home. “*C:\Users\****YOUR\_USER\_NAME***”. This makes it easy when we wish to change directories (**CD**) in the terminal. I will assume for this guide that you have done the same, so if you have unpacked the *“model”* folder to a different directory make sure you reflect those changes when copying the rest of the terminal commands.

So your *“model”* folder should now be at the directory *“C:\Users\****YOUR\_USER\_NAME*** *\model”*.

If you closed the command prompt terminal, then reopen it and make sure to activate the virtual environment with **activate capstone**

With the environment activated **CD** to the model folder. Then enter the following command to install the model dependencies.

**pip install -r requirements.txt**

When that is done **CD** into the *“youtube”* folder which should be at *“C:\Users\****YOUR\_USER\_NAME****\model\youtube”*. Now we want to download a test video to run through the model to test the installation.

From this directory we can enter **youtube-dl URL\_OF\_YOUTUBE\_VIDEO -f mp4** into the terminal and it will download an mp4 version of the YouTube video at the link you provide. Since we want to test a video that we know will work we’ll use one of the videos the model was tested on. So let’s download a dodgeball video from “https://youtu.be/--0edUL8zmA”. Type in:

**youtube-dl https://youtu.be/--0edUL8zmA -f mp4**

The video “*Dodgeball from Above.mp4*” should have been downloaded into the *“youtube”* folder. Usually it is a good idea to rename the videos into something easier to type into the terminal for the next step. I renamed the video to *“Dodgeball.mp4”.*

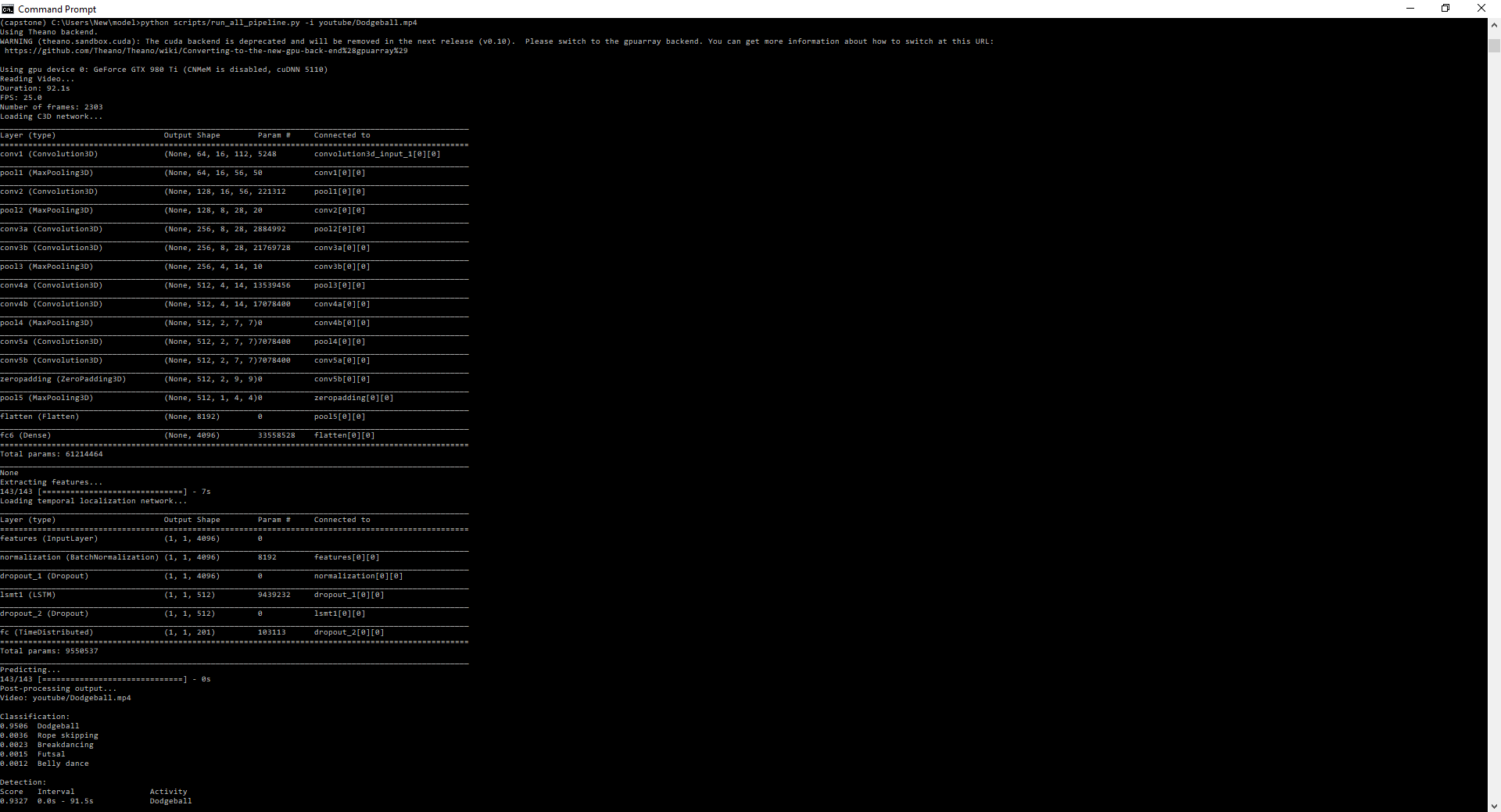
**NOTE: You can find a list of YouTube URLs of videos used to train or test the model if you go to the *“C:\Users\YOUR\_USER\_NAME\model\dataset”* directory and look in *“videos.json”* or *“videos\_id.lst”* files. Use those as a reference of you want to try some other videos.**

**5. Analysing a video.**

**CD** back to the *“model”* folder and type the following:

**python scripts/run\_all\_pipeline.py -i youtube/Dodgeball.mp4**

You should get the following terminal readout (again, you may have to zoom in):



If you didn’t get this then something must have gone wrong during one of the previous steps. Remember that your virtual environment needs to be activated. It needs to be activated again every time you open the terminal.

**6. Guide References:**

Original Model available from:

<https://imatge-upc.github.io/activitynet-2016-cvprw/>

Original Setup guide for the Activity Dataset Model:

<https://github.com/imatge-upc/activitynet-2016-cvprw/blob/master/misc/step_by_step_guide.md>