These are the set of commands used in all three devices:

* Windows
  + Jupyter notebook and python 3.6 was used to train models from scratch using keras framework with tensorflow backend (CUDA enabled)
  + The model generated can be found in the model folder of this repository
  + SSH was done to the linux machine running ubuntu to execute git related commands
* Linux(Ubuntu)
  + The vitis-ai-tutorials was cloned from the following git repository to convert keras model into elf models. <https://github.com/Xilinx/Vitis-AI-Tutorials.git>
  + Commands run inside the repository:
    - ./docker\_run.sh xilinx/vitis-ai:latest (to start the docker)
    - source ./0\_setenv.sh (initializing many of the parameters for the specific model)
    - source ./2\_keras2tf.sh (starting the keras to tensorflow conversion)
    - source ./4\_quant.sh (quantization of the images placed inside the calib\_images folder)
      * A custom datagen.py script was written to convert the calib\_images into proper format for the model input shape
    - source ./6\_compile.sh ( compiling the model with path to the dcf and arch json file inside the DOUCZDX8G path)
  + The above command must have generated an elf file if executed properly.
* Ultra96V2 (Petalinux)
  + The DPU runner was used to utilise the built elf model. Threads were used for the static version and no threads were utilised for the live feed version
  + The target board being Ultra96V2 obtained the code via git function. The original git version of the model can be found in this repo with all the commits: <https://github.com/version0chiro/Xillinx_ULTRA96_BIRDS>
  + There are two version of the code present in the git repo, one running on static images while the other working on live feed taking input frames directly from the Logitech webcam attached to the usb port of ultra96.
    - Running the static code: python3 recognize\_birds.py -i {path to the image directory}
      * The code was executed in a batch format
    - Running the live feed version : python3 bird\_recognition\_live\_feed.py.py