SVILUPPO DI UN SISTEMA DI PERSON DETECTION TRAMITE INTEL MOVIDIUS

Installazione del software di Person Detection

Marco Pettorali
marcopettorali1@gmail.com

Sommario

Installazione del software di Person Detection

Premesse:

Il materiale necessario per l'installazione del software è il seguente:

- Neural Compute Stick Movidius™ v.1 di Intel®
- Raspberry Pi 3B+
- webcam USB

E' importante sottolineare che questa trattazione è valida **soltanto** per la prima versione di Movidius, altresì chiamata NCS1. Lo SDK che verrà utilizzato, infatti, non supporta la nuova versione del Neural Compute Stick, NCS2.

Si suppone di avere precedentemente installato sul Raspberry Pi la versione Lite del sistema operativo Raspbian Stretch, scaricabile da questo indirizzo:

https://www.raspberrypi.org/downloads/raspbian/

In questo modo avremo un'installazione di base del sistema operativo, che non include applicazioni e pacchetti non utili per la realizzazione di un riconoscitore di presenza umana. Per collegarsi al Raspberry Pi utilizzeremo il protocollo ssh; per configurare Raspberry Pi come server ssh basterà creare un file vuoto, senza estensione, dal nome 'ssh' nella scheda SD, insieme agli altri file del sistema operativo: questo file verrà letto da Raspbian, ignorato il contenuto, e successivamente eliminato. Questa operazione andrà svolta soltanto al primo avvio del Raspberry Pi, dopo il quale non sarà più necessaria.

Prerequisiti:

Avremo inizialmente bisogno di installare alcuni pacchetti e dipendenze sui quali i programmi di interfaccia con Movidius si appoggiano. Dopo aver aggiornato le package lists con il comando `sudo apt-get update`, possiamo installare i tool 'git' e 'pip3', che ci torneranno estremamente utili nelle fasi successive dell'installazione. Si noti che non sarà necessario installare prima i pacchetti di python3, perché questo è già integrato in Raspbian Stretch.

```
sudo apt-get update
sudo apt-get install git
sudo apt-get install python3-pip
```

Installazione del Neural Compute SDK:

Prima di proseguire, è consigliabile aumentare la dimensione dello swap file del Raspberry Pi. Per farlo è sufficiente modificare la variabile CONF_SWAPFILE del file /etc/dphys-swapfile, impostandola al valore 1024.

```
echo CONF_SWAPSIZE=1024 > dphys-swapfile-enlarged
sudo mv /etc/dphys-swapfile /etc/dphys-swapfile-original
sudo mv dphys-swapfile-enlarged /etc/dphys-swapfile
sudo /etc/init.d/dphys-swapfile restart
```

Adesso proseguiamo con l'installazione di ncsdk2, l'SDK di Movidius. Notare l'opzione per cambiare branch dalla repository Github in fase di cloning `-b ncsdk2`: sul branch master della repository, attualmente (febbraio '19) c'è ncsdk, ovvero la versione precedente e deprecata dell'SDK. E' già stato

annunciato dalla Intel sul file README.md della repository, che a breve ncsdk2 verrà spostato sul master branch, quindi il comando di cloning riportato qua sotto potrebbe non essere più valido.

Dopo l'operazione di cloning, assicurarsi che lo Movidius sia correttamente collegato in una porta USB del Raspberry Pi, e proseguire con il comando `make install`.

```
cd ~
sudo git clone -b ncsdk2 https://github.com/movidius/ncsdk.git
cd ncsdk
make install
```

Per controllare che l'installazione di ncsdk2 sia avvenuta con successo, eseguiamo il programma d'esempio fornito all'interno della repository, senza scollegare Movidius dal Raspberry Pi.

```
cd examples/apps/hello_ncs_py
make run
```

Se ci sono stati errori durante l'esecuzione, si consiglia di rimuovere la cartella 'ncsdk' in cui il comando git clona la repository, di ripetere il comando di cloning e provare a reinstallare nuovamente ncsdk2.

Installazione di openCV:

La repository su Github contiene anche uno script per l'installazione di openCV, un framework molto utile per la raccolta, l'elaborazione e la visualizzazione delle immagini, e viene utilizzato da molte applicazioni di object detection. Per installarle basta dare il comando `./install-opencv.sh`. L'installazione può durare molto tempo, attorno alle due ore.

```
cd ~/ncsdk
./install-opencv.sh
```

Dall'installazione di openCV mancano, però, alcune dipendenze fondamentali senza le quali python3 non riconoscerà le librerie di openCV. Questo sembra essere un problema di compatibilità con Raspberry Pi, in quanto molti utenti Raspberry lamentano problemi di mancanza di dipendenze di vario genere, problemi non sperimentati dagli altri utenti di openCV.

Infatti, se all'interno della console di python3, si proverà a dare il comando 'import cv2', questo puntualmente terminerà con errore, indicando ogni volta quale file sta cercando di caricare e non trova. Interpretando questi messaggi di errore, e svolgendo alcune ricerche, ho individuato i seguenti pacchetti mancanti:

```
pip3 install opencv-python
sudo apt-get install libgstreamer1.0-0
sudo apt-get install libqtgui4
sudo apt-get install libqt4-test
pip3 install -U numpy=1.15.4
```

Adesso possiamo finalmente testare l'installazione di openCV e delle librerie di python3 ad esso collegate. Aprendo la console di python con il comando `python3` e inserendo il comando `import cv`, non dovremmo vedere adesso nessun errore, e l'importazione termina con successo.

Se, invece, si fosse verificato un errore, interpretare il messaggio nell'ultima riga dello stack trace per cercare di capire quale è il file mancante. Successivamente fare una ricerca su Internet per capire in quale pacchetto è contenuto il file mancante. Consiglio vivamente di usare l'elenco ufficiale dei pacchetti stable di Debian, al link https://packages.debian.org/stable. In alto a destra, e a sinistra della barra di ricerca, selezionare la voce 'package contents', quindi digitare nella barra di ricerca il nome del file ricercato. Premendo il pulsante 'Search', apparirà una lista di pacchetti nel quale si trova il file in questione. Sceglierne uno, ponendo attenzione sull'architettura del pacchetto: quella di Raspberry Pi 3B+ è 'arm64'. Se la query dovesse terminare con esito nullo, provare nei packages in fase di testing, al link https://packages.debian.org/testing.

Se questo metodo dovesse fallire, fare una ricerca tra i forum ufficiali di Movidius (https://ncsforum.movidius.com/), openCV (http://answers.opencv.org/questions/), e Raspberry Pi (https://www.raspberrypi.org/forums/), oppure postare un quesito sui suddetti forum. Provare anche su altri forum non ufficiali (uno su tutti Stack Overflow http://stackoverflow.com) in cui contattare altri utenti che hanno affrontato lo stesso problema o problemi simili.

Se in precedenza è stato modificato il file /etc/dphys-swapfile, è consigliabile riportare la variabile CONF_SWAPFILE reimpostandola al valore di default 100.

```
sudo mv /etc/dphys-swapfile /etc/dphys-swapfile-enlarged
sudo mv /etc/dphys-swapfile-original /etc/dphys-swapfile
sudo /etc/init.d/dphys-swapfile restart
```

Programmi di Person Detection:

Andremo ad utilizzare due software, indipendenti l'uno dall'altro, entrambe basate su MobileNet SSD, un'implementazione delle reti MobileNets sviluppata da Google, compatibile con il framework Caffe.

Le reti MobileNets (presentate nell'articolo https://arxiv.org/pdf/1704.04861.pdf), sono una classe di reti neurali progettate per essere estremamente veloci, precise e a basso consumo, indicate per applicazioni embedded quali il Raspberry Pi.

Caffe (http://caffe.berkeleyvision.org/) è, invece, un framework sviluppato dalla Berkeley AI Research, che permette la descrizione, l'ottimizzazione, il training e il testing di reti neurali.

Il primo che installeremo è NCS-Pi-Stream, realizzato e caricato su GitHub da HanYangZhao. E' un software di Live Object Detection, la SSD MobileNet che usa è stata allenata su un set di immagini in modo tale che categorizzi gli oggetti individuati nelle 20 categorie proposte per la prima volta nella VOC2007 Challenge, organizzata da PASCAL Visual Object Classes (http://host.robots.ox.ac.uk/pascal/VOC/), raccolte in quattro macro-classi: Person (che è quella che ci interessa maggiormente), Animal, Vehicle, Indoor.

NCS-Pi-Stream sfrutta openCV per l'acquisizione delle immagini dalla webcam e per la successiva visualizzazione di riquadri (box) attorno agli oggetti trovati, sfruttando le informazioni in output fornite da MobileNets e integra un semplice web server che visualizza in diretta le immagini analizzate e elaborate in formato .mjpg, come sequenza di immagini jpg.

Il secondo è invece il programma 'SSD_MobileNet/run.py', incluso nella repository GitHub 'ncappzoo' che contiene tutti gli esempi ufficiali (sviluppati da Movidius Ltd.) per la dimostrazione del Neural Compute Stick. In particolare, questo programma funge da dimostrazione di MobileNet SSD e, a differenza di NCS-Pi-

Stream, non offre un'interfaccia web, ma stampa il risultato a video sull'emulatore di terminale, e non scatta un'immagine da una webcam per analizzarla, ma di base sfrutta un immagine di esempio contenuta nella repository Ncappzoo. Ugualmente a NCS-Pi-Stream, include una rete allenata secondo le categorie del VOC2007.

Ho modificato quest' ultimo esempio realizzandogli una semplice interfaccia web e adattandolo per l'acquisizione di frame provenienti dalla webcam collegata al Raspberry Pi.

Installazione di NCS-Pi-Stream:

Andiamo dunque a creare una cartella in cui posizionare i programmi di Person Detection:

```
cd ~
mkdir -p workspace
cd workspace
```

Cloniamo da GitHub la repository di NCS-Pi-Stream:

```
git clone https://github.com/HanYangZhao/NCS-Pi-Stream.git
cd NCS-Pi-Stream/models
```

Poiché il file graph, utilizzato da ncsdk2 per caricare la rete neurale all'interno di Movidius, contenuto nella repository sembra corrotto e non riconosciuto da Movidius, è necessario ricompilarlo tramite il comando 'mvNCCompile' fornito nel ncsdk. Questa operazione è possibile perché l'autore della repository, ha incluso in una cartella i file .prototxt e .caffemodel per la ricompilazione del file graph. In particolare il file .prototxt rappresenta una descrizione della rete neurale MobileNet conforme al framework Caffe, mentre il file .caffemodel contiene i pesi della rete ottenuti dopo le iterazioni di training eseguite con Caffe stesso.

```
mvNCCompile MobileNetSSD_deploy.prototxt -s 12 -w MobileNetSSD_deploy.caffemodel
mv graph ../graph/mobilenetgraph
```

Avviando il programma con il comando `python3 streamer_ncs.py`, e visitando la pagina http://YOUR_PI_IP:8080/cam.mjpg, dove al posto di YOUR_PI_IP bisogna inserire l'indirizzo IP al quale è collegato il Raspberry Pi, si potranno vedere in diretta le immagini elaborate.

Installazione di SSD_MobileNet

Per l'installazione di questo esempio non è necessario scaricare tutta la repository ncappzoo, dunque scaricheremo solo i file necessari per l'installazione:

```
cd ~/workspace
mkdir SSD_MobileNet
cd SSD_MobileNet
wget
https://raw.githubusercontent.com/movidius/ncappzoo/master/caffe/SSD_MobileNet/Makefile
wget
https://raw.githubusercontent.com/movidius/ncappzoo/master/caffe/SSD_MobileNet/deploy_prototxt_update.patch
wget
https://raw.githubusercontent.com/movidius/ncappzoo/master/caffe/SSD_MobileNet/labels.txt
```

```
wget
https://raw.githubusercontent.com/movidius/ncappzoo/master/caffe/SSD_MobileNet/run
.py
wget
https://raw.githubusercontent.com/movidius/ncappzoo/master/caffe/SSD_MobileNet/REA
DME.md
make all
```

Prima di eseguire il programma con il comando `python3 run.py`, è necessario aprire il file run.py con un editor e andare a modificare la variabile globale in cui è memorizzato il percorso del file da analizzare, con il percorso della propria immagine da elaborare. Questo perché il percorso di default fa riferimento ad un percorso che, non avendo installato la versione completa di ncappzoo, non sarà ovviamente presente nel file system del Raspberry Pi.