1. **INTRODUZIONE**

In un mondo dove i grandi colossi dell’informatica hanno ormai monopolizzato enormi quantità di dati visuali, come ad esempio immagini e video, raccogliendone numerose serie dai loro consumatori, e utilizzandoli per “allenare” i propri algoritmi, accaparrandosi quindi un grosso vantaggio su tutti i competitors, risulta indispensabile l’utilizzo dei dati sintetici per riuscire a sviluppare un sistema di computer vision (disciplina che studia come i computer sono in grado di riprodurre funzioni dell’apparato visivo dell’uomo) per un singolo individuo. Tramite questi dati è infatti possibile per chiunque riuscire ad addestrare i propri algoritmi in modo efficiente per svariati settori, come ad esempio la robotica, la sorveglianza e i sistemi per la guida autonoma. Capiamo meglio cosa si intende per dati sintetici.

I synthetic data sono dati che vengono generati dal computer in grado di simulare dati reali. Quindi essi non vengono creati manualmente ma tramite una simulazione del computer. Sono utilizzati per far capire alla nostra macchina come comportarsi in determinate situazioni. Durante la fase di allenamento risultano molto importanti le etichette, o meglio tag, perché in questo modo il computer è in grado attribuire un significato ai dati visuali che riceve. In passato questo tipo di dati non erano molto utilizzati nonostante avessero il vantaggio di essere gratuiti, perché c’era la preoccupazione che essi riuscissero a fornire ottimi risultati nelle simulazioni, ma non altrettanto una volta inseriti in un processo naturale.

Un altro aspetto da non tralasciare riguarda il fatto che è molto difficile descrivere il mondo tramite un piccolo campione di immagini, inoltre tag con alta qualità richiedono tempo e denaro per essere acquisiti.

Tramite i dati sintetici si è in grado di descrivere un aspetto piccolo ma di molta rilevanza nel mondo, con dettagli perfetti. Perciò si può affermare che i dati sintetici hanno ridotto il vantaggio delle grandi potenze dell’intelligenza artificiale, dando modo a piccoli team di competere grazie ai loro costi ridotti e alla elevata velocità di sviluppo che garantiscono.

**1.1 GitHub**

Durante il lavoro svolto sono stati utilizzati alcuni tools per lo sviluppo del sistema di hand detection. Innanzitutto c’era bisogno di uno strumento che permettesse di condividere con gli altri colleghi di tesi il lavoro svolto dal singolo, in modo che tutti poi potessero usufruirne e implementare correzioni e modifiche. Di conseguenza è stato utilizzato GitHub, un servizio di hosting che permette di effettuare le

operazioni sopra descritte tramite la linea di comando Git CMD.

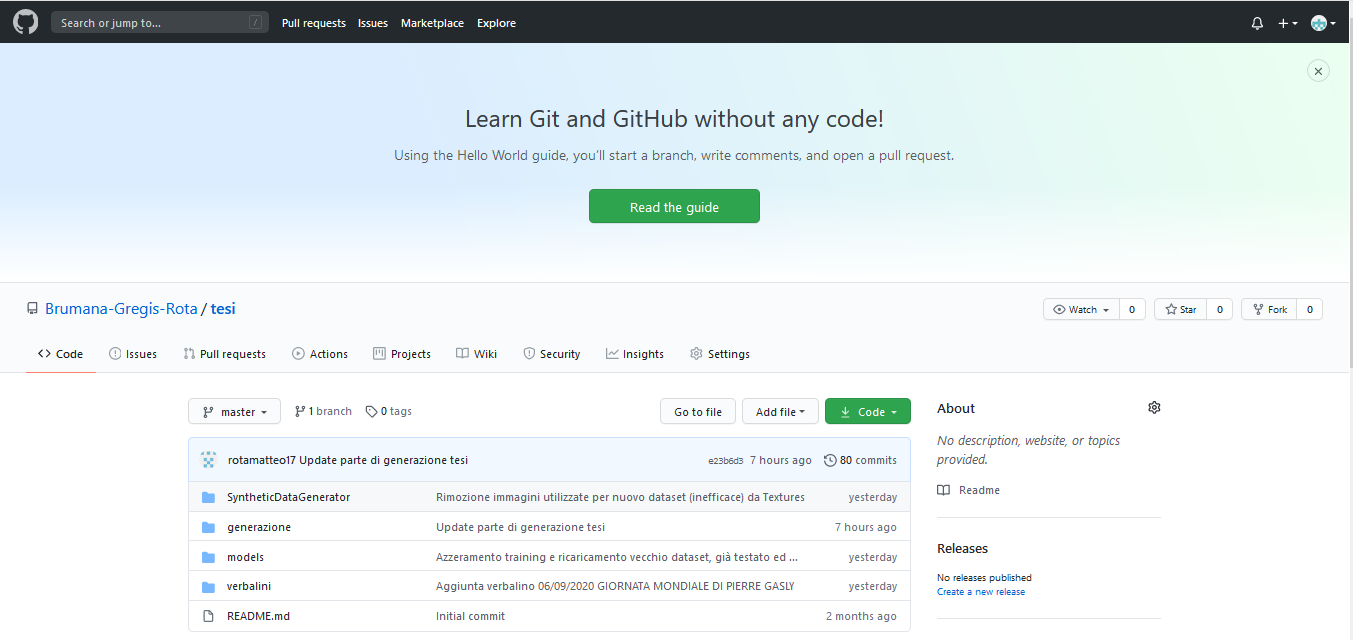


Figura 1.1 – pagina principale di GitHub

Fondamentalmente i comandi utilizzati sono quelli relativi al caricamento di un nuovo file all’interno della cartella condivisa (git add, git commit -m, git push), e quelli per scaricare contenuti nuovi aggiunti dagli altri membri del gruppo (git pull).

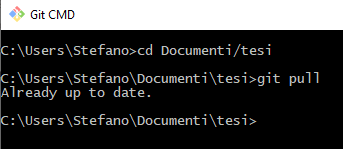


Figura 1.2 – esempio di comando pull tramite git CMD per il download dei nuovi contenuti

Nella figura 1.2 viene mostrato il download dei nuovi contenuti, effettuato accedendo dalla linea di comando di Git nell’apposita cartella tramite il comando *cd,* segue poi l’aggiornamento dei file tramite il comando *git pull* (al

momento della cattura dell’immagine non vi erano nuovi file nella cartella).

Nell’immagine 1.3 si può notare invece l’upload di un nuovo file che avviene, sempre posizionandosi inizialmente nella cartella condivisa, con 3 comandi: il primo è git add*,* utilizzato per selezionare i file da caricare. Successivamente vengono aggiunti i file della index (quelli selezionati precedentemente) all’head tramite il comando git commit -m “commento” che, inoltre, da la possibilità di inserire un messaggio descrittivo. Ed infine vengono caricati i file nella cartella tramite git push.

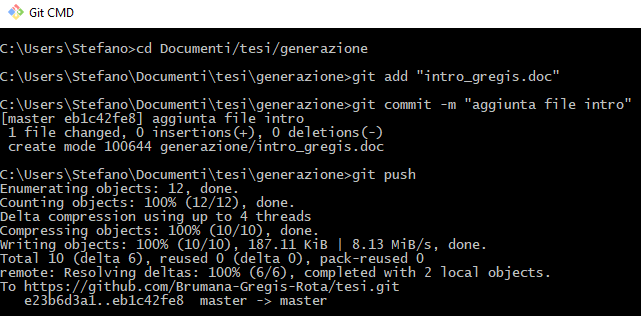


Figura 1.3 – esempio di caricamento di nuovi file nella cartella condivisa

Dopo aver analizzato e introdotto le principali fuzioni di GitHub, nei paragrafi successivi verranno trattati i tools utilizzati per creare il nostro sistema di hand detection.

**1.2 Unity e Fatkun**

Unity 3D è un engine necessario per lo sviluppo di videogiochi, il quale può essere utilizzato per la generazione delle immagini che successivamente alleneranno la rete.

La scelta di questo “motore grafico” è dovuta non tanto per le sue performance o per la qualità del rendering, infatti non è il migliore in circolazione da questo punto di vista, ma soprattutto per la sua grande flessibilità e semplicità. Con pochi e non molto complessi passaggi vengono ricreate numerose immagini di train contenenti diversi ambienti e luoghi con la presenza, al loro interno, delle mani.

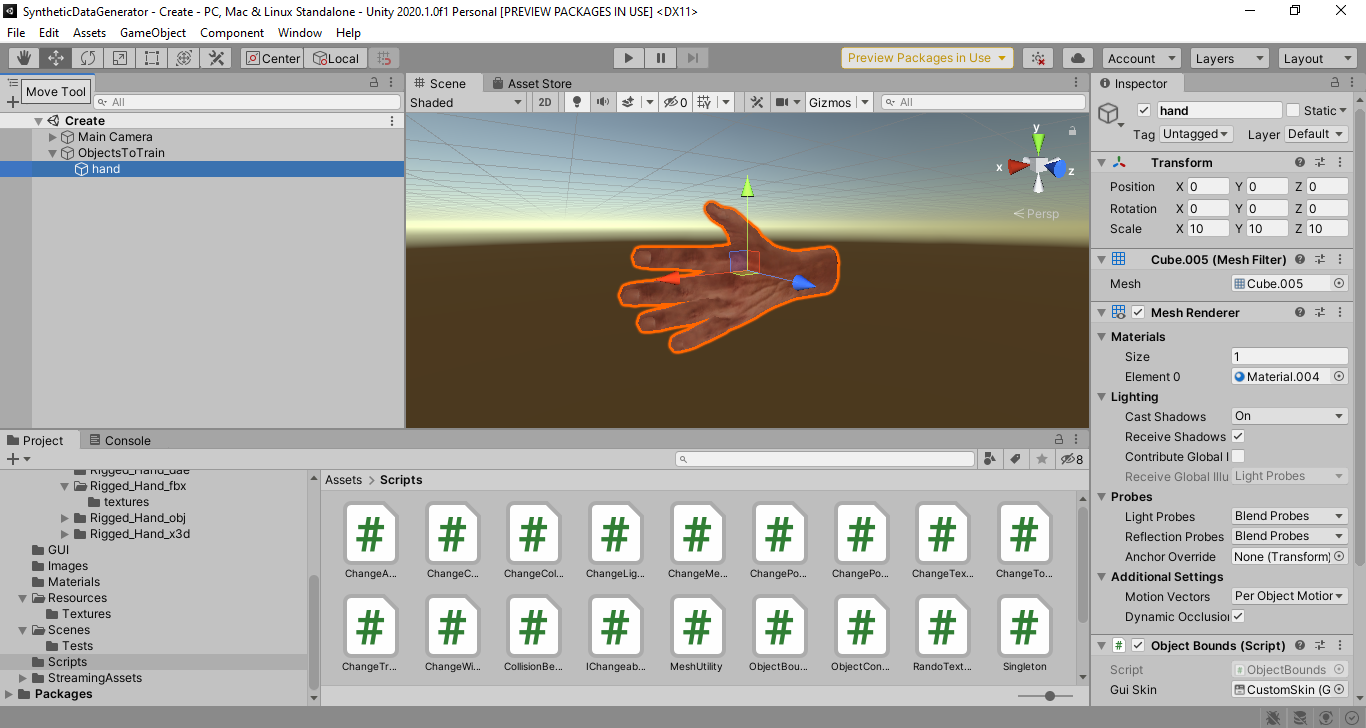


Figura 1.4 – schermata principale di Unity 3D

Fondamentali, per quanto riguarda questo programma, sono le *scene*, ossia la rappresentazione dell’ambiente (ad esempio un giardino con una mano), che permettono di visualizzare gli oggetti in 3D.

All’interno di essa ci si può muovere e cambiare il punto di vista, tenendo premuto il tasto destro del mouse e spostandosi nella direzione desiderata. In questo modo è come se si girasse la telecamera nelle 4 direzioni disponibili su di un piano immaginario. Inoltre è possibile effettuare lo zoom per avvicinare o allontanare la telecamere. Ovviamente sono consentite altre operazioni, ma ai fini del nostro progetto, queste sono le principali cui è stato fatto ricorso.

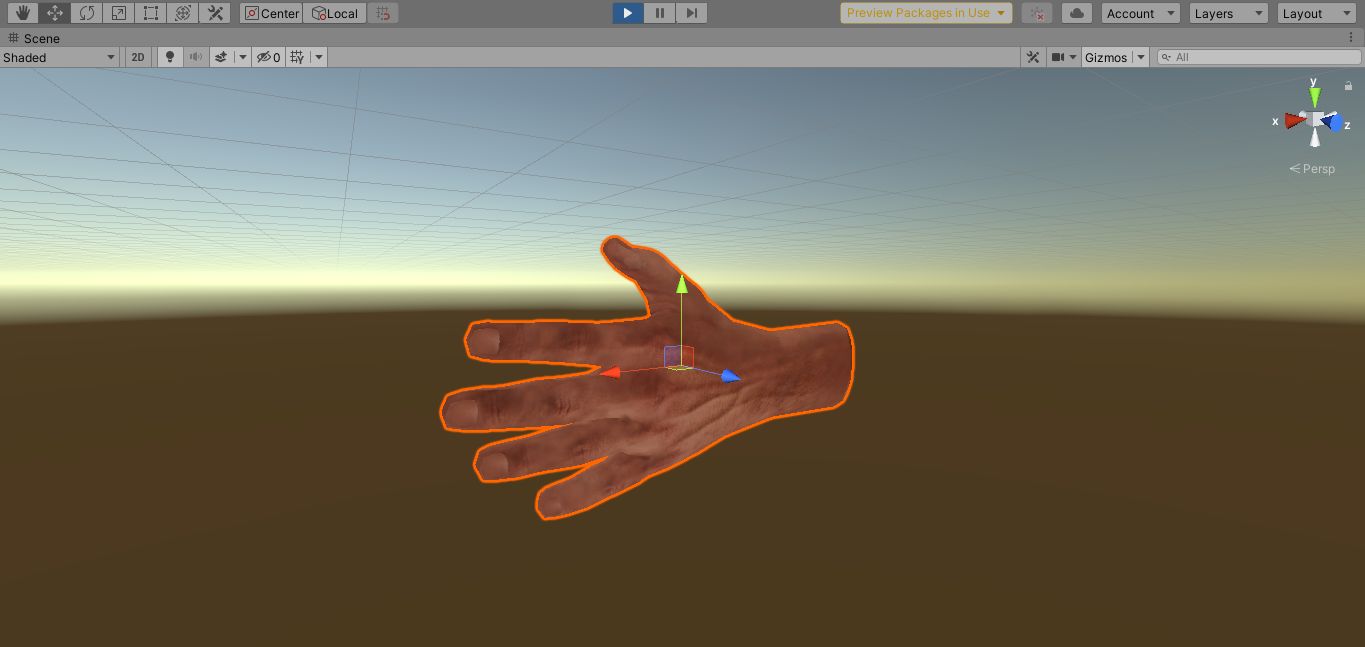


Figura 1.5 – esempio di scena

Aspetto importante sono i file .meta, creati da Unity nel momento in cui importiamo l’oggetto nel workspace. Dettagliano come avverrà l’importazione del file a cui fanno riferimento.

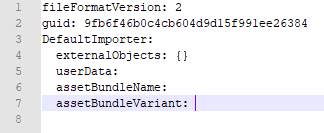


Figura 1.6 – esempio di file .meta

Simultaneamente a Unity è stato utilizzato Fatkun Batch DownloadImage, un’estensione di Google Chrome che consente di scaricare le numerose immagini sulle schede aperte durante il download.

**1.3 Tensorflow e Tensorflow Object Detection API**

Arrivati a questo punto occorre qualcosa per creare e allenare la nostra rete. Quindi si è installato Tensorflow, una libreria open source creata per il calcolo numerico che utilizza grafici di flusso di dati. Si può considerare Tensorflow come una macchina che parla in tensori, ovvero una serie di numeri, sottoforma di vettori e matrici, che può essere utilizzata per l’apprendimento della rete.

Le API all’interno di questa libreria sono sviluppate in Python il quale consente di implementare e richiamare più velocemente le librerie di Tensorflow.

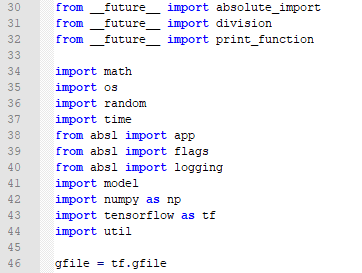


Figura 1.7 – esempio di file di Unity scritto in linguaggio Python con importazione di Tensorflow

Di particolare interesse per quanto riguarda il nostro progetto è Tensorflow Object Detection API, un framework open source che si basa su TensorFlow ed è la parte che effettivamente facilita lo sviluppo e l’allenamento.

Nella nostra rete è stata implementata la versione 1.15, provando anche a migrare il tutto alla versione 2.0. Purtroppo questo non è stato possibile in quanto ssd\_moblienet\_v2 non è supportato, quindi si è mantenuta la versione utilizzata precedentemente.