ABSTRACT:

La proposta iniziale di stage era quella di analizzare alcuni framework per il riconoscimento dai FAKE images attraverso l’utilizzo di NN e Deep Learning. L’idea era alquanto interessante.

Per procedere con l’esperienza si è deciso di andare a capire il funzionamento di quello che c’è a monte delle FAKE-IMAGES, cioè la loro creazione.

Si è passati quindi ad un progetto che andasse a studiare il funzionamento e la qualità dei risultati di Frame Work per i virtual Try-On (VTO). La generazione, quindi, di immagini false, artefatte, create attraverso l’utilizzo di codice basato su Reti neurali e Machine Learning, ad hoc per l’ambito della moda e non solo.

Il punto di partenza è stato il codice VITON (portato al CVPR 2018). Una rete per il try-on virtuale image-based (partendo da un dataset di immagini 2D con l’obbiettivo di creare modelli 3D). Codice che sarà la base di partenza e di confronto con gli altri analizzati durante l’esperienza.

Proseguendo poi al ricercare un concorrente, sempre basato su VITON, per permettere dei confronti prestazionali, nel caso analizzato è stato utilizzato Virtual Try-on with Detail Carving (VTODC). Framework portato al CVPR 2019 e realizzato dalla JDAI CV, con sede a Pechino, che introduce la possibilità di scegliere anche con quale posa generare il modello finale, oltre all’abito e al soggetto.

Infine si è deciso di produrre una demo che permettesse di scegliere una combinazione di soggetti/abiti da un dataset limitato ed eseguire la sostituzione del vestiario con l’opzione di poter introdurre modifiche riguardanti la suddivisione fra i vari segmenti che compongono soggetto di base: vestito, braccia/maniche, pantaloni, viso e capelli.

OBBIETTIVI;

L'obbiettivo che ci si è posti come cardine dell'esperienza di tirocinio è stato quello di valutare i risultati, quindi le prestazioni, di più codici che eseguono virtual try-on con base (VITON). Le richieste erano quelle di capire quando i due codici confrontati producevano output validi e quando invece presentavano difficoltà nel produrli.

Si è quindi proceduto ad analizzare diversi codici basati sul predecessore VITON e scegliere quelli che potevano avere incrementi sostanziali e visibili nelle prestazioni e nella qualità/affidabilità dei risultati.

Durante l'esperienza si è inoltre deciso di provare ad aggiungere ulteriori obiettivi ossia: Produrre un piccolo applicativo "CANVAS" che permettesse la modifica del label (suddivisione delle componenti del soggetto finale: viso, vestito, braccia, pantaloni, capelli) e applicarlo poi ad una demo "PITON" che permettesse ad un utente di generare differenti outfit, attraverso l'utilizzo di un dataset di campioni ridotto, potendo inoltre utilizzare CANVAS per modificare alcune caratteristiche degli abiti/soggetto.

COS'E' UN VIRTUAL TRY-ON:

Partiamo spiegando cos'è un virtual try-on. È una modalità alternativa con la quale un futuro cliente di un negozio di abbigliamento o di gioielli potrà provare i prodotti dei negozi in modo virtuale, senza dover fisicamente provarli in camerino. Quindi attraverso l'utilizzo di un dispositivo dotato di fotocamera, l'utente può valutare se comprare o meno un capo provandolo virtualmente, attraverso l'utilizzo di realtà aumentata.

Questo approccio, già presente da anni sul mercato ma migliorato mediante l'utilizzo di machine learning

e modelling 3D, permette di provare un capo prima di acquistarlo in qualsiasi luogo, senza l'"obbligo" di andare in negozio, di provare contemporaneamente più capi, valutando anche i vari outfit e soprattutto è time-saving.

Nel caso preso in esame si è operato su codici che prevedevano l'utilizzo di dataset di immagini prese da

un catalogo di abbigliamento di un negozio di e-commerce, quindi non attraverso l'utilizzo di modelling 3D e risultati real-time applicati su video, ma sulla creazione di outfit alternativi a quelli originali utilizzando le

coppie modello target e vestito target.

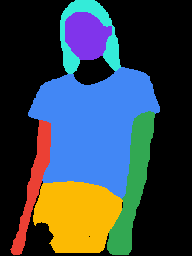
PYVTON:

L’applicativo che si è deciso di produrre doveva soddisfare due requisiti principali:

1. Permettere all’utente la possibilità di scegliere la coppia Modello-Abito
2. Modifica delle segmentazioni relative alle componenti che formavano l’immagine di base:

Ciano= capelli, Violetto = viso, Azzurro = Abito, Giallo = Pantaloni, Arancio/Rosso = Braccio DX

Verde = Braccio SX



È stato scelto come Framework per la sostituzione dell’abito il sorgente ACGPN, portato all’CVPR nell’edizione 2020.

Sistema realizzato nell’ambiente di sviluppo CUDA, quindi l’architettura hardware creata da NVIDIA per l’elaborazione parallela. Per eseguire questo codice è stato necessario l’utilizzo di macchine più performanti ed è stato possibile grazie all’accesso alle macchine dell’Università di Verona, su cui si è lavorato tramite SSH.

PYVTON è formato da due componenti principali: la prima è il wrapper che contiene CANVAS (app per la modifica dei label) e lo script per la selezione delle immagini sorgenti dal dataset di cui è dotato il programma, la seconda invece, è l’eseguibile test.py contenuto nella sezione di ACGPN dedicata alle fasi di testing, ossia quella di Inference.

Tralasciando le parti già presenti nella repository iniziale di ACGPN, il codice è stato scritto completamente in Python. Il codice di PYVTON è poi stato portato all’interno dell’ambiente di lavoro di ACPGN in modo da aver accesso diretto a test.py e soprattutto al Dataset.

Per quanto riguarda quest’ultima componente, il dataset utilizzato nella demo prodotta è un sottoinsieme di quello utilizzato per ACGPN e VITON. Per le fasi di test e debugging si è optato per un dataset della dimensione di 5 modelli e 5 abiti scelti in modo casuale da quello originale, selezionando quelle immagini che non portassero a problemi di Missing keypoints o deformazioni varie nel risultato (stessi problemi riscontrati nei test di VITON).

INTERFACCIA:

Quello che l’utente si trova davanti è un’interfaccia minimale e alquanto spartana, non era necessario aggiungere tanti dettagli, ma rendere il più possibile leggero l’applicativo dovendolo eseguire via SSH.

L’interfaccia è così suddivisa:

(1) Anteprima dei Soggetti utilizzabili come base per il VTO.

(2) Relative codifiche delle immagini dei soggetti.

(3) Menu a tendina per la selezione del soggetto.

(4) Tasto avviare l’applicativo per la modifica delle suddivisioni del soggetto.

(5) Anteprima degli Abiti utilizzabili come target per il VTO.

(6) Come sopra, relative codifiche delle immagini degli abiti.

(7) Menu a tendina per la selezione degli abiti.

(8) Avvio del file di test.py per la produzione del risultato.

Passando all’interfaccia della mini applicazione per la modifica delle segmentazioni “CANVAS” abbiamo:

(9) Menù a tendina per la selezione di import/export dell’immagine da modificare.

(10) Menu a tendina per la selezione del colore del pennello in base alla parte della segmentazione che si vuole modificare.

(11) Cursore per la selezione della dimensione del pennello.

(12) Tasto reset, per rimuovere l’immagine che si stava modificando.

(13) Piano di disegno su cui si può modificare l’immagine sorgente.