**Documentazione per la tesi**

**Introduzione**

Lo studio dello yoga e delle varie pose, chiamate “asanas” ha una lunga e ricca storia, che abbraccia centinaia di anni. Ad oggi lo yoga è riconosciuto per i suoi innumerevoli benefici, come il miglioramento della salute fisica, la riduzione dello stress e delle ansie e il miglioramento della concentrazione e della consapevolezza di se. Di conseguenza, la pratica dello yoga è diventata sempre più popolare sia tra i più giovani che tra i più anziani, con milioni di persone in tutto il mondo che la incorporano nella loro routine quotidiana. Tuttavia, la corretta esecuzione delle posizioni yoga è fondamentale per evitare lesioni e massimizzare i benefici della pratica.

Garantire una corretta esecuzione non è però un compito facile, difatti ci sono molti fattori, come l’anatomia individuale, la flessibilità, la forza, che possono incidere sull’esecuzione della posa. Pertanto valutare accuratamente la propria forma può risultare complicato sia per i principianti sia per i più esperti. Ed è qui che nasce l’esigenza di un sistema che permetta la classificazione e la correzione dell’esecuzione di pose di yoga.

Inoltre dopo la pandemia COVID-19 l’aumento della domanda per sistemi di yoga di questo tipo e più in generale di sistemi per il fitness, ha reso questo problema ancor più rilevante.

Quindi questo tipo di sistema fornisce all’utente una totale autonomia garantendogli una maggior sicurezza ed efficacia nell’esecuzione degli esercizi. Oltre a poter essere utilizzato in maniera autonoma dall’utente, questo tipo di sistemi potrebbe essere usato a sostegno di istruttori, sia che questi svolgano le lezioni all’interno di una palestra, sia che questi svolgano le lezioni online su piattaforme dedicate, che come detto precedentemente, a causa della pandemia sono molto diffuse, fornendogli tutte le informazioni per poter seguire in maniera oculata l’esecuzione della posa da parte di tutti i suoi allievi e non solo una parte di questi.

Questa tesi quindi avrà come obbiettivo quello di realizzare il sitstema sopra descritto, e per farlo verranno utilizzate tecniche di computer vision e di machine learning, con un focus sul deep learning.

**Capitolo 1**

Come anticipato il nostro sistema farà uso di tecniche di computer vision machine learning e deep learning, pertanto in questa sezione andremo ad esplorare questi argomenti ad alto livello per dare al lettore una panoramica generale di quello che verrà approfondito nei capitoli successivi.

**Artificial Intelligence**

Cosa è l’intelligenza artificiale? Esistono diverse definizioni di intelligenza artificiale, che si sono susseguite nel corso del tempo, tra le più accreditate vi è la definizione di John McCarthy, il quale offre la seguente definizione nel paper [1], rilasciato nel 2004:

*" It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable."*

Definizione che trova riscontro nelle principali applicazioni dell’Intelligenza Artificiale come speech recognition, customer service, computer vision, reccomendation engines, automated stock trading, ossia applicazioni che normalmente richiedono intelligenza umana ma non solo.

L’AI può essere divisa in 2 categorie[2]:

* **Weak AI:** la Weak AI, chiamata anche Narrow AI o Artificial Narrow Intelligence (ANI), è un'IA addestrata e focalizzata per eseguire compiti specifici. L'intelligenza artificiale debole è sicuramente il tipo di AI ad oggi più diffuso. 'Stretto/Narrow' potrebbe essere una descrizione più accurata per questo tipo di IA in quanto è tutt'altro che debole; infatti si trova in alcune delle applicazioni ad oggi più robuste, come Siri di Apple, Alexa di Amazon, IBM Watson e veicoli autonomi
* **Strong AI:** la Strong AI è composta dll’ Intelligenza Generale Artificiale (AGI) e Super Intelligenza Artificiale (ASI). L'intelligenza artificiale generale (AGI), o IA generale, è una forma teorica di IA in cui una macchina avrebbe un'intelligenza pari a quella umana; avrebbe una coscienza autocosciente che ha la capacità di risolvere problemi, imparare e pianificare il futuro. La super intelligenza artificiale (ASI), nota anche come super intelligenza, supererebbe l'intelligenza e l'abilità del cervello umano. Sebbene l'IA forte sia ancora interamente teorica senza esempi pratici in uso oggi, ciò non significa che i ricercatori di intelligenza artificiale non stiano anche esplorando il suo sviluppo. Nel frattempo, i migliori esempi di ASI potrebbero provenire dalla fantascienza, come HAL, l'assistente informatico sovrumano e canaglia in 2001: Odissea nello spazio.

Oltre a poter dividere l’AI in queste due categorie, possiamo individuare anche 2 approcci/paradigmi:

* Symbol AI
* Machine Learning

Symbol AI implica lo sviluppo di regole, fatti e modelli espliciti(KB) per rappresentare la conoscenza ed eseguire compiti. L'apprendimento automatico, d'altra parte, comporta lo sviluppo di algoritmi in grado di apprendere automaticamente dai dati; in particolare ai fini di questa tesi ci concentreremo sul machine learning.

L'intelligenza artificiale ha quindi il potenziale per trasformare un'ampia gamma di settori, dalla sanità ai trasporti fino ad arrivare alla finanza e al commercio al dettaglio. Ad esempio, nel settore sanitario, gli algoritmi di intelligenza artificiale vengono utilizzati per migliorare l'accuratezza delle diagnosi mediche e per sviluppare piani di trattamento personalizzati. Nei trasporti, l'intelligenza artificiale viene utilizzata per sviluppare veicoli autonomi e migliorare la gestione del traffico.

Tuttavia, lo sviluppo e la diffusione dell'IA sollevano anche una serie di importanti preoccupazioni etiche e sociali. Ad esempio, il crescente utilizzo dell'IA nel processo decisionale solleva importanti questioni sulla responsabilità e sulla trasparenza. Inoltre, il potenziale dell'intelligenza artificiale di automatizzare i lavori e spostare i lavoratori solleva importanti domande sul futuro del lavoro e sulla disparità di reddito. Oltre a questi problemi ne sorgono molti altri di più varia natura, raccolti all’interno dello studio[3]

**Machine Learning**

Come anticipato precedentemente una sottocategoria di intelligenza artificiale è il machine learning, il quale in accordo con uno dei suoi pionieri,  Arthur Samuel, è definito come : “the field of study that gives computers the ability to learn without explicitly being programmed.”[4]

Difatti il ML si pone come obbiettivo quello di realizzare sistemi che possono apprendere dai dati e fare previsioni o prendere decisioni senza che questi siano stati esplicitamente programmati per eseguire compiti specifici. In altri termini il ML è il processo di training(addestramento) di un pezzo di software, chiamato model(modello), che permette di fare predizioni utili a partire dai dati. Quindi un modello altro non rappresenta che la relazione matematica tra gli elementi dei dati che un sistema di ML usa per fare predizioni.

Due dei task più comuni che vengono risolti con sistemi basati sull’apprendimento supervisionato sono i task di classificazione e regressione. Un modello di regressione predice un valore numerico mentre i modelli di classificazione predicono la probabilità(la likelihood) con la quale il sample preso sotto esame appartenga ad una data categoria. Inoltre i modelli di classificazione possono effettuare una classificazione binaria o una classificazione multi-classe.

Gli algoritmi di machine learning possono essere divisi in 3 categorie a seconda della modalità con cui avviene la fase di apprendimento[5]:

* **Supervised learning**
* **Unsupervised learning**
* **Reinforcement learning**

Nei modelli di apprendimento supervisionato le previsioni vengono fatte dopo aver visto molti dati con le risposte corrette e quindi aver scoperto le connessioni tra gli elementi nei dati che producono le risposte corrette. Da un punto di vista più tecnico avremo che il nostro modello verrà addestrato su una serie di dati dove oltre alle features di input sono definite le features target.

Due dei task più comuni che vengono risolti con sistemi basati sull’apprendimento supervisionato sono i task di classificazione e regressione. Un modello di regressione predice un valore numerico mentre i modelli di classificazione predicono la probabilità(la likelihood) con la quale il sample preso sotto esame appartenga ad una data categoria. Inoltre i modelli di classificazione possono effettuare una classificazione binaria o una classificazione multi-classe.

Nei modelli di apprendimento senza supervisione le previsioni vengono fatte ricevendo dati che non contengono risposte corrette. L'obiettivo di un modello di apprendimento non supervisionato è identificare pattern significativi tra i dati. In altre parole, il modello non ha suggerimenti su come classificare ogni dato, ma deve invece inferire le proprie regole. Da un punto di vista più tecnico avremo che il nostro modello verrà addestrato su una serie di dati dove sono definite solo le features di input e non le features target.

Una delle tecniche più usate nell’apprendimento non supervisionato è il clustering, dove a partire dai dati vengono creati dei cluster che andranno a dividere i nostri samples.

Nei modelli di apprendimento per rinforzo le previsioni vengono effettuate ottenendo ricompense o penalità in base alle azioni eseguite all'interno di un ambiente. Un sistema di apprendimento per rinforzo genera una politica/policy che definisce la migliore strategia per ottenere il maggior numero di ricompense.

L'apprendimento per rinforzo viene utilizzato per addestrare i robot a svolgere compiti, come camminare per una stanza, e programmi software come AlphaGo per giocare al gioco del Go.

Nonostante le potenzialità del ML anche questi tipi di sistemi hanno delle importanti limitazioni e problematiche da affrontare. Una di queste è l’overfitting, che si verifica quando l'algoritmo apprende troppo bene i dati di addestramento, al punto da non essere più in grado di generalizzare su nuovi dati, non ancora osservati. L'overfitting può però essere mitigato attraverso tecniche come la regolarizzazione, la cross validation e l'arresto anticipato.

Un’altra problematica da affrontare è il problema del bias nell’apprendimento. Gli algoritmi di machine learning possono essere “distorti”/ “influenzati” se i dati di addestramento contengono degli errori sistematici o se l’algoritmo è stato progettato con determinati bias. Questo può comportare ad ottenere risultati non corretti o discriminatori pertanto bisogna stare molto attenti a questi problemi quando si decide di distribuire/mettere in produzione un algoritmo di ML.

**Deep Learning**

Un sottocampo del ML è il Deep learning, il quale si concentra sullo sviluppo di algoritmi e modelli ispirati alla struttura e al funzionamento del cervello. A differenza degli algoritmi di machine learning gli algoritmi di deep learning hanno però bisogno di una mole di tanto molto più elevata.

Gli algoritmi di deep learning sono costituiti da più layer di neuroni artificiali, che sono collegati da connessioni pesate/ponderate. L'input al primo livello sono i dati grezzi, come un'immagine o una frase. Dopo di che l'output di ogni livello viene passato come input al livello successivo, finché il livello finale non produce la previsione o decisione finale. Durante la fase di training l'algoritmo regola i pesi delle connessioni per ridurre al minimo l'errore tra gli output previsti e quelli effettivi.

Uno dei principali vantaggi nell’uso di algoritmi di DL rispetto a algoritmi di ML è che il deep learning elimina parte del processo di pre-processing dei dati tipicamente coinvolto nel machine learning. Infatti questi algoritmi possono importare ed elaborare dati non strutturati(raw-data), come testo e immagini, e automatizzare l'estrazione delle funzionalità, eliminando la dipendenza da degli esperti umani. Ad esempio, supponiamo di avere una serie di foto di diversi animali domestici e di volerli classificare in base a 'gatto', 'cane', 'criceto', eccetera. Gli algoritmi di deep learning possono determinare quali caratteristiche (ad esempio le orecchie) sono più importanti per distinguere un animale da un altro. Nell'apprendimento automatico, questa gerarchia di funzionalità viene stabilita manualmente da un esperto umano.[6]

Tuttavia, il deep learning oltre a presentare gli stessi problemi del machine learning presenta anche il problema dell'interpretabilità, che si riferisce alla difficoltà di capire perché un modello di deep learning abbia fatto una particolare previsione o decisione. Questa può essere una sfida importante per applicazioni come la diagnosi medica, dove è importante comprendere le basi per una diagnosi.

**Computer Vision**

La Computer vision come anticipato è una branchia dell'intelligenza artificiale che si occupa dell'interpretazione e dell'analisi delle informazioni visive. Implica lo sviluppo di algoritmi, modelli e tecniche per consentire ai computer di comprendere, interpretare e manipolare le informazioni visive dal mondo, come immagini e video. Il campo della computer vision è motivato dal desiderio di replicare la percezione visiva umana e di fornire ai computer la capacità di eseguire attività che normalmente richiederebbero l'intelligenza visiva umana, come riconoscere oggetti, rilevare schemi e prendere decisioni basate su dati visivi. Inoltre grazie ai progressi dell'intelligenza artificiale e alle innovazioni nel deep learning e nelle reti neurali, il campo ha potuto fare grandi passi avanti negli ultimi anni ed è stato in grado di superare gli umani in alcuni compiti relativi al rilevamento e all'etichettatura di oggetti.

Uno dei fattori principali ad oggi che ha portano alla crescita della computer vision è la quantità di dati che generiamo e che poi usiamo nei nostri modelli, si stima infatti che oggi vengano condivise più di 3 miliardi di immagini e 720000 ore di video ogni giorno[7].

Immagini che vengono altro non sono delle matrici bidimensionale di pixel, ognuno con la propria codifica a seconda dello spazio di colore usato.

Ad esempio nel seguente esempio[8] è stata usata la scala di grigi, pertanto ogni pixel sarà rappresentato da un numero a 8-bit il cui range va da [0,255], dove 0 corrisponde a nero e 255 a bianco.

Immagine che contiene testo, interni, piastrellato, tegola

Descrizione generata automaticamente

Oltre alla scala di grigi potremo avere come modelli colore anche quello RGB e HSV. Sia nel primo caso che nel secondo avremo che ogni pixel sarà rappresentato da 3 byte, ma mentre nel primo caso ogni byte codifica rispettivamente il colore, red, green e blue nel secondo caso 1 byte viene usato per la tonalità, 1 byte per la saturazione e 1 byte per la luminosità.

**Capitolo 3**

**Strumenti utilizzati**

**Capitolo 2**

**Classificazione**

All’interno di questo capitolo verrà descritta l’intera pipeline di lavoro che ha portato al risultato ottenuto, a partire dai dati a nostra disposizione fino ad arrivare ai modelli usati per la classificazione. In particolare, ogni scelta presa verrà spiegata in maniera approfondita con apposite regressioni di natura teorica dove necessario.

**Dataset**

**Organizzazione e Analisi esplorativa dei Dati**

Per la realizzazione di questo sistema è stato utilizzato il seguente dataset[9], che fa parte di una collezione Open Source realizzata nello studio proposto [10]. Questo dataset si compone di 88 video, rappresentanti 15 diversi individui(5 femmine e 10 maschi), di diversa età, che svolgono 6 diverse pose di yoga:

* Cobra (Bhujangasana)
* Tree (Vrikshasana)
* Mountain (Tadasana)
* Lotus (Padmasana)
* Triangle (Trikonasana)
* Corpse (Shavasana)

Ogni video ha una durata che si aggira tra i 30 e i 60 secondi per un totale di 1 ora 6 minuti e 5 secondi. Tutti i video sono stati registrati a 30 fps con una qualità pari a 1366x768. Inoltre tutti i video sono stati registrati all’interno di una stanza ad una distanza di 4 metri dalla camera.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Yoga Pose | #Persone | #Video |
| 1 | Cobra | 15 | 16 |
| 2 | Lotus | 14 | 14 |
| 3 | Corpse | 15 | 15 |
| 4 | Mountain | 15 | 15 |
| 5 | Triangle | 13 | 13 |
| 6 | Tree | 15 | 15 |
|  | Numero totale di video |  | 88 |

 Immagine che contiene persona, interni, pavimento, ragazza

Descrizione generata automaticamente 

Prima di passare effettivamente al pre-processing dei nostri dati i video sono stati ripartiti in maniera manuale in 6 diverse cartelle rappresentanti le 6 diverse pose

Inoltre per garantire una maggior reliability sono stati presi alcuni accorgimenti sulla struttura e l’organizzazione del progetto e sono stati realizzati una serie di script che permettono di lavorare più facilmente con i dati a nostra disposizione.

**Human Pose Estimation**

Come appena detto nel paragrafo precedente il nostro dataset si compone di una serie di video in cui vengono eseguite diverse pose ma, per poter effettuare una classificazione dei video, dobbiamo affrontare il problema della **Human Pose Estimation.**

La Human Pose Estimation (HPE) è un task della computer vision che si concentra sull’identificazione della posizione di un corpo umano all’interno di una scena specifica. Più precisamente è un modo per catturare un insieme di coordinate per ogni articolazione (braccio, testa, busto, ecc.) , che è nota come key-point e che vengono usate per descrivere una posa di una persona. La connessione tra due punti è nota come coppia.

La connessione formata tra i punti deve però essere significativa, il che significa che non tutti i punti possono formare una coppia. Fin dall'inizio, l'obiettivo di HPE è quello di formare una rappresentazione 2D o 3D di un modello del corpo umano.

Questi modelli quindi sono sostanzialmente delle mappature dei joints del corpo umano, sia che questo stia fermo, sia che questo sia in movimento.

Esistono tre tipi di modelli per la modellazione del corpo umano[12]:

* **Kinematic model** o **Skeleton-based model**

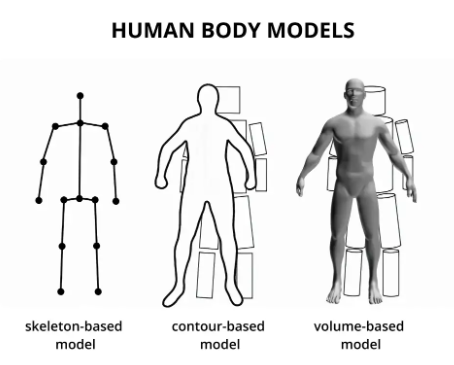
viene utilizzato per la stima della posa 2D e per la stima della posa 3D. Questo modello di corpo umano flessibile e intuitivo include una serie di posizioni articolari, joints e orientamenti degli arti per rappresentare la struttura del corpo umano. Pertanto, i modelli di stima della posa dello scheletro vengono utilizzati per catturare le relazioni tra le diverse parti del corpo. Tuttavia, i modelli cinematici sono limitati nella rappresentazione delle informazioni sulla texture o sulla forma .

* **Planar model o Countourn-based model**

viene utilizzato per la stima della posa 2D. I modelli planari sono usati per rappresentare l'aspetto e la forma di un corpo umano. Di solito, le parti del corpo sono rappresentate da più rettangoli che si avvicinano ai contorni del corpo umano.

* **Volumetric model o Volume-based model**

viene utilizzato per la stima della posa 3D. Esistono diversi modelli di corpo umano 3D popolari utilizzati per la stima della posa umana 3D basata sull'apprendimento profondo per il recupero della mesh umana 3D.

[11]

Concentrandoci sugli skeleton based-models, essendo questi di nostro interesse, abbiamo che i modelli possono essere suddivisi in 2 categorie a seconda dell’approccio adottato per determinare lo scheletro della persona:

1. **Top down:** vengono localizzati prima le persone all’interno della scena e poi sono individuate le parti del corpo
2. **Botton Up:** vengono prima localizzate le parti del corpo e a partire da quelle si individua l’intero corpo

Dopo una serie di ricerche e sfruttando anche lo studio [13] in cui vengono confrontati i punti di forza di vari modelli si è deciso di utilizzare il modello Move-Net.

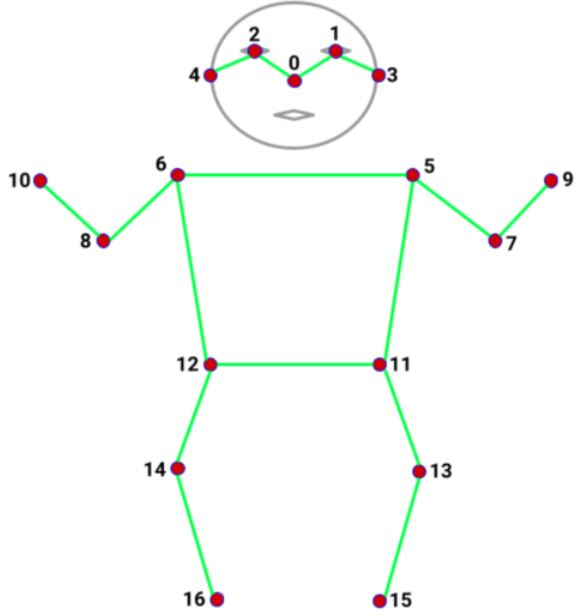
Si è deciso di usare Move-Net oltre che per le sue peculiarità quali la velocità nella rilevazione dei punti, la capacità di individuare i punti per più persone .

**Move-Net**

MoveNet è un modello ultra veloce e preciso che rileva 17 punti chiave di un corpo. Il modello è offerto su TF Hub con due varianti, note come Lightning e Thunder.

Lightning è destinato ad applicazioni critiche per la latenza, mentre Thunder è destinato ad applicazioni che richiedono un'elevata precisione.

Entrambi i modelli funzionano più velocemente del tempo reale (oltre 30 FPS) sulla maggior parte dei desktop, laptop e telefoni moderni, il che si rivela fondamentale per le applicazioni di fitness, salute e benessere dal vivo.

**#TO DO scrivere la parte di movenet [14]**

**Creazione del dataset**

Come indicato all’inizio di questo capitolo per effettuare la classificazione come dataset non utilizzerò direttamente i video o i frames ma andrò a costruire un nuovo dataset a partire dai landmarks individuati e restituiti da Move-Net.

Pertanto definisco una funzione che effettui l’estrazione dei landmarks a partire dall’output del modello appena descritto. Avremo quindi che per ogni frame verrà restituito un tensore che verrà trasformato dapprima in un array bidimensionale che avrà 17 righe corrispondenti ai 17 joints e 3 colonne corrispondenti a x, y e confidence e poi verrà reso flatten ossia ad una dimensione. Tale vettore sarà ritornato dalla funzione. Quindi saremo passati da una shape:

1. (1, 1, 17, 3)
2. (17, 3)
3. (51,)

Si noti che il vettore è stato reso flatten poiché le LSTM(un modello di deep learning per la classificazione) lavorano con questo formato di input.

Per garantire l’organicità del progetto e mantenere l’organizzazione congrua all’organizzazione fino ad ora descritta, definisco una funzione che per ogni video crei una rispettiva cartella, nominata con “nome video + \_landmarks”, all’interno della quale veranno salvati i keypoint raccolti per ogni frame.

Dopo aver creato le cartelle definisco la funzione che andrà effettivamente a creare il nostro dataset, infatti questa funzione per ogni frame di ogni video andrà ad estrarre, tramite la funzione prima descritta, i keypoints che verranno salvati come numpy array all’interno della rispettiva cartelle.

*N.B da questo momento in poi quando parliamo di video o frames facciamo sempre riferimento ad array di 51 points o a gruppi di questi nel caso di video.*

**Divisione del dataset**

Affinché possa creare dei set di traning e test i samples del dataset devono essere etichettati, ossia ogni sample deve essere associata la categoria di appartenza.

Pertanto definisco un dizionario che associa ad ogni posa una label, che andrà ad indicare la classe di appartenza, quindi in questo momento sto definendo il dominio della nostra feature target.

{'Cobra': 0, 'Corpse': 1, 'Lotus': 2, 'Mountain': 3, 'Tree': 4, 'Triangle': 5}

Si noti come creando questo dizionario si sia passati da una rappresentazione categoriche ad una numerica senza sfruttare il label encoding fornito dalla libreria Sci-Kit Learn.

Nel momento dell’assegnazione delle label si è deciso di non assegnare la label al singolo video composto da centinaia se non migliaia di frame ma si è deciso di dividere i video in subset di 45 frames, quindi in video da 1,5 secondi, ed assegnare ad ognuno di questi una label. In questo modo quindi avremo un dataset composto non da soli 88 video ma da 2599 video, ognuno dei quali composto da 45 frames.

Arrivati a questo punto effettuo la divisione del nostro set di dati in set di training e set di test, andando a comporre rispettivamente il set di training con l’80% dei dati, mentre il set di test con il 20% dei dati. Essendo comunque il nostro dataset un dataset bilanciato, avendo circa gli stessi samples per ogni categoria, si è deciso di non splittare in modo casuale ma di utilizzare il parametro stratify nella funzione di splitting, in modo tale da mantenere le proporzioni del vettore assegnato; in questo caso il vettore è il vettore colonna rappresentante le features target.

Come risultato avremo il seguente numero di video:

* 520 video per il test
* 2079 video per il train

BIBLIOGRAFIA

1. <https://www-formal.stanford.edu/jmc/whatisai.pdf>
2. <https://www.ibm.com/topics/artificial-intelligence>
3. <https://link.springer.com/chapter/10.1007/978-3-030-69978-9_4>
4. <http://infolab.stanford.edu/pub/voy/museum/samuel.html>
5. <https://developers.google.com/machine-learning/intro-to-ml/what-is-ml>
6. <https://www.ibm.com/topics/deep-learning>
7. <https://www.qut.edu.au/study/business/insights/3.2-billion-images-and-720000-hours-of-video-are-shared-online-daily.-can-you-sort-real-from-fake>
8. <https://xd.adobe.com/ideas/principles/emerging-technology/what-is-computer-vision-how-does-it-work/>
9. <https://archive.org/details/YogaVidCollected>
10. <https://link.springer.com/article/10.1007/s00521-019-04232-7#Sec2>
11. <https://arxiv.org/pdf/2006.01423.pdf>
12. <https://viso.ai/deep-learning/pose-estimation-ultimate-overview/>
13. <https://maureentkt.medium.com/selecting-your-2d-real-time-pose-estimation-models-7d0777bf935f>
14. <https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html>