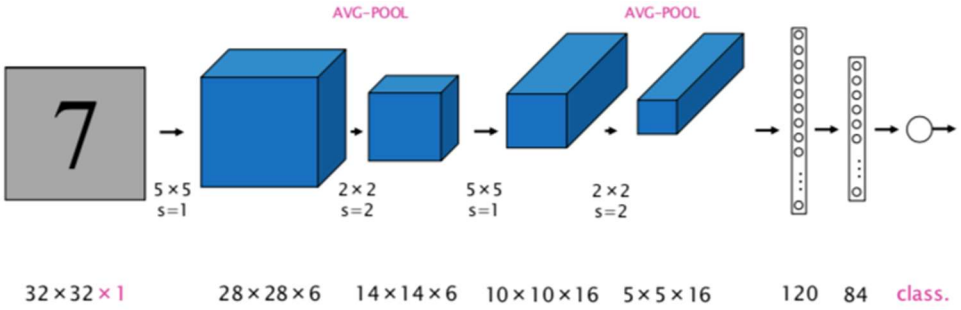
10 - CONVOLUTIONAL NEURAL NETWORKS: CASI DI STUDIO E TASKS

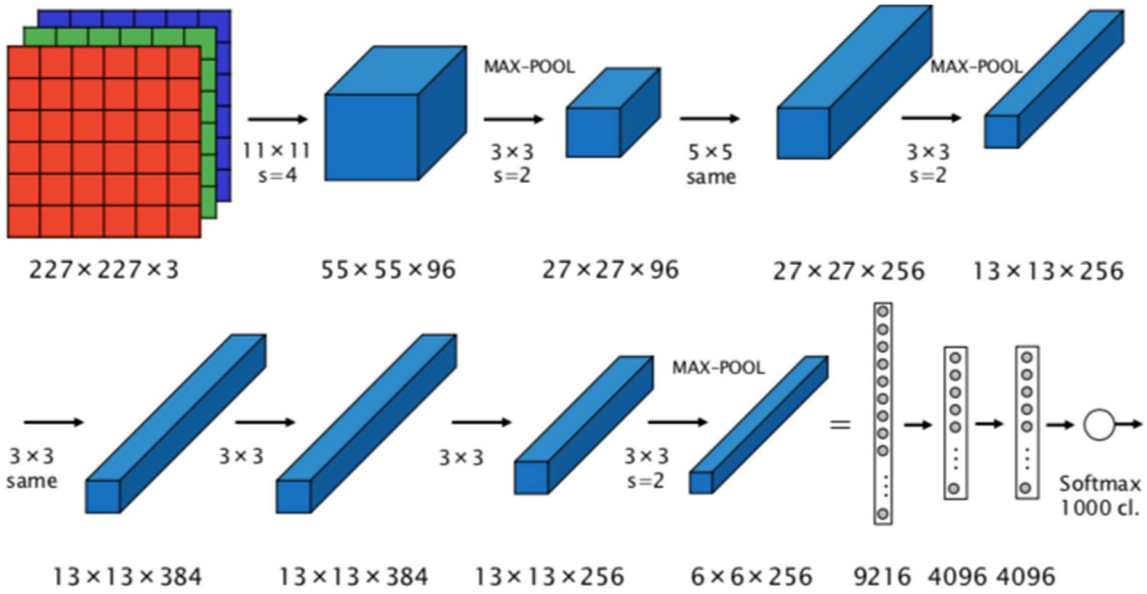
LeNet-5

Rete del 1998 per riconoscere numeri scritti a mano.



AlexNet

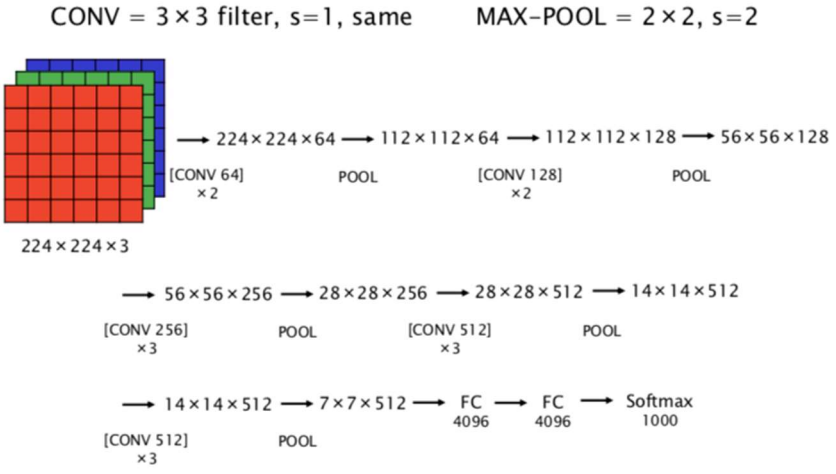
Sviluppata nel 2012. Si parte con un livello convolutivo a cui si applicano filtri 11x11 con stride=4, il numero di filtri non è riportato ma lo si capisce dalla profondità dell'output della convoluzione che è 96. L'input si riduce velocemente in quanto non c'è padding.
Poi viene applicato un max-pooling ottenendo il primo livello (conv+max.pool).
Questo tipo di livello viene ripetuto e poi si fanno una serie di convoluzioni same riducendo il numero di filtri e un max-pool finale.
Si fa infine l'unroll connettendo dei livelli fuly connected per finire in un softmax che riconosce 1000 classi.



Ha un sacco di parametri (circa 60 milioni), di hidden unit e funziona bene con i suoi hyper-parametri. È la rete che ha convinto sull'utilizzo delle deep network.
Funziona bene con fine-tuning.

VGG-16

Semplifica la filosofia di AlexNet: anziché utilizzare un grosso numero di hyper-parametri si vogliono usare solo blocchi conv e max-pool.



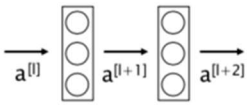
Questa ha il doppio dei parametri di AlexNet circa 138 milioni.
Ciò che ha reso famosa questa rete è la "semplicità", rispetto ad AlexNet ci sono meno hyperparametri.

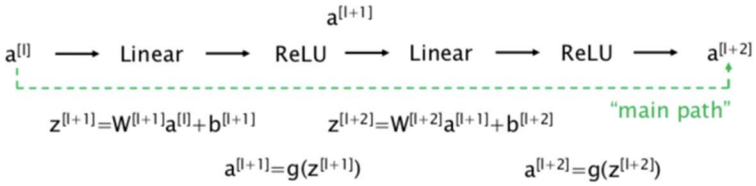
Residual block

In una rete neurale il primo livello cosa fa?

- Prendiamo gli input del livello precedente
- moltiplico per i parametri del livello attuale e sommo il bias.
- Ottengo z che è la parte lineare.
- Applico una non linearità g (esmepio ReLU) ottenendo le attivazioni di quel livello.

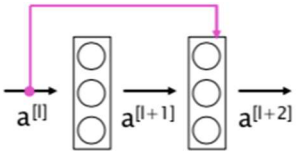
Per il livello successivo si può fare la stessa cosa.

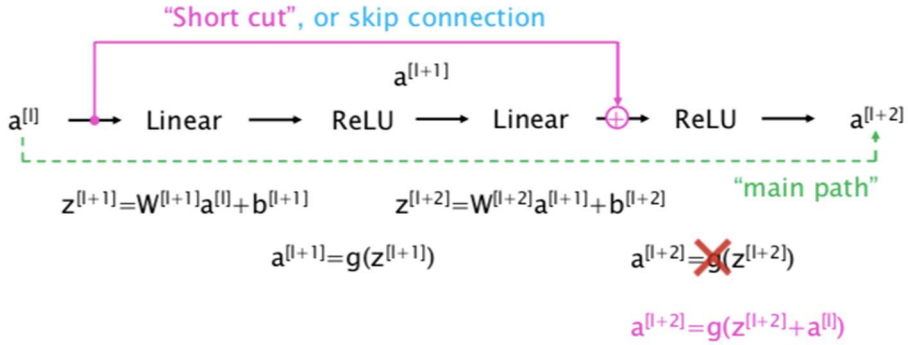




Tutto questo viene chiamato il **main path**.

L'idea che sta dietro al residual block è questa: è stato ideato un possibile **shortcut/(skip connection)** dove si voleva provare a portare più avanti l'input del primo livello andandolo a sommare dopo la linearità del secondo livello, saltando di fatto un livello. Con questo meccanismo l'ultimo step in cui calcolo le attivazioni le calcolerò sulla z del secondo livello più gli input del primo.

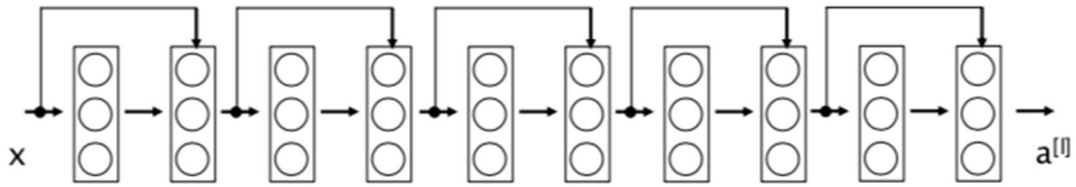




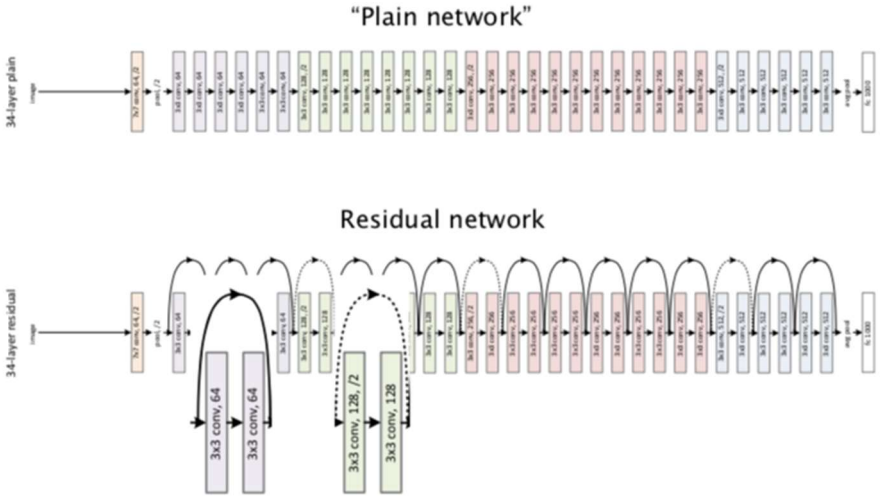
Ipotizzando di trovarci in una rete grande, dopo che sarà applicata la regolarizzazione, la z del secondo livello sarà un valore vicino allo 0. Immaginando gli input del primo livello positivi si ha che la g avrebbe solo la parte positiva e quindi è come se la g non ci fosse perché la ReLU si attiva con i valori positivi → l'uscita del secondo livello corrisponde agli input del primo livello. Sembra che quel blocco riesca ad imparare molto bene la funziona identità e può permettermi di imparare qualcosa di più, di più interessante.

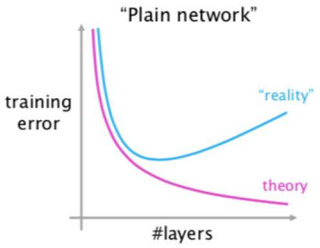
Residual Network (ResNet)

Viene presa una “plain network” a cui vengono aggiunte le skip connection.

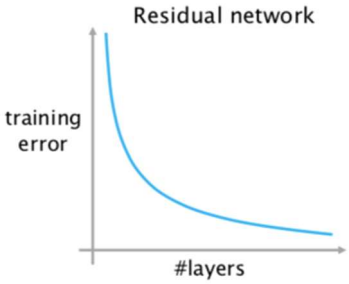


In una rete plain standard confrontando il training error all’aumentare del numero di layer ci si aspetta che l’errore diminuisca, in realtà non accade così perché più la rete è profonda più diventa complicato convergere.





Con la residual network invece accade come ci si aspetta nella teoria.



Dal lavoro originale dei ricercatori, si evince che copn una rete residuale si può continuare a far crescedre il numero di livelli, continuando ad abbassare l’errore. A fermarci è la potenza di calcolo o il numero di dati che si hanno a disposizione.

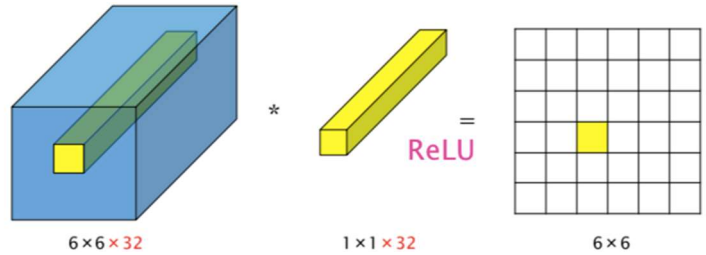
Ci si sono alcuni accorgimenti adottati ad esempio nelle convoluzioni dove viene fatto un uso esasperato del padding per poter mantenere le dimensioni per le skip connections.

Convoluzioni 1x1

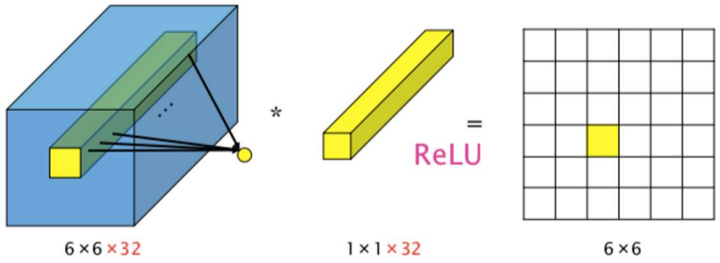
Utili per tantissime finalità

Se prendiamo una matrice bidimensionale, fare una convoluzione con un filtro 1x1 si riduce a fare il prodotto della matrice per uno scalare.

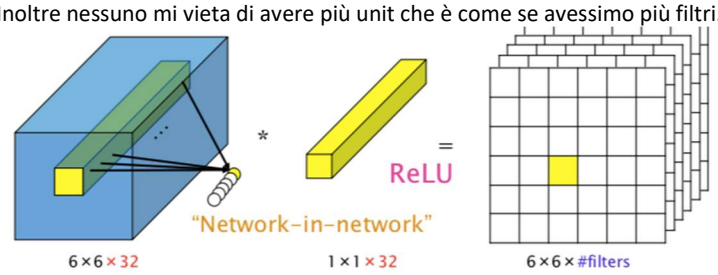
Se noi avessimo una dimensione maggiore useremmo un filtro grande 1x1x(profondità della matrice). In questo caso l'output fa scomparire la profondità della matrice facendo **moltiplicare per il parametro e poi sommare la singola** striscia all'unico valore della cella di output.



È un tipo di operazione non lineare in pratica.
Questo è come se avessi una unit che prende i 32 input, li combina tra loro, applica una non linearità e mi da un'uscita. (considera bias)
Questo approccio viene chiamato network-in-network, perché è **come se stessimo creando una rete neurale all'interno della nostra rete**.

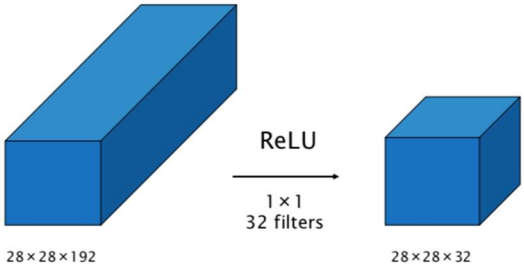


Inoltre nessuno mi vieta di avere più unit che è come se avessimo più filtri. Ognuno di loro produce uno dei foglietti.



Usare le convoluzioni 1x1 → si comporta come una rete neurole

Un utilizzo può essere quello di ridurre le dimensioni della profondità ed è dimostrato che non crea svantaggio nella rete.

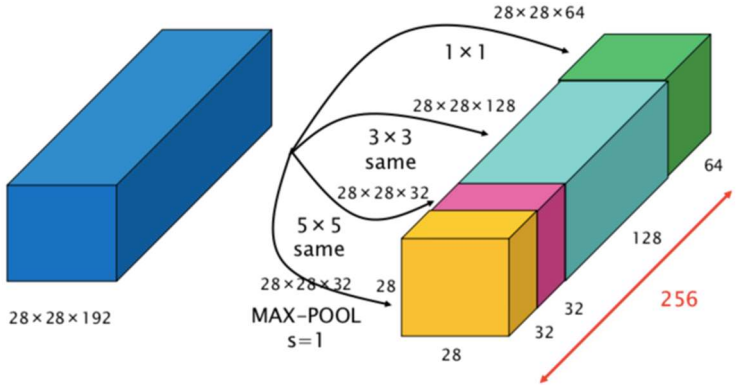


Questo **fa risparmiare calcoli**.
Addirittura, si può pensare di applicare lo stesso numero di filtri 1x1 della profondità dell'input per aggiungere una non linearità, quindi per aggiungere un passaggio di aggiornamento.
Può essere utilizzata per cambiare in maniera controllata la terza dimensione dei volumi.

Inception

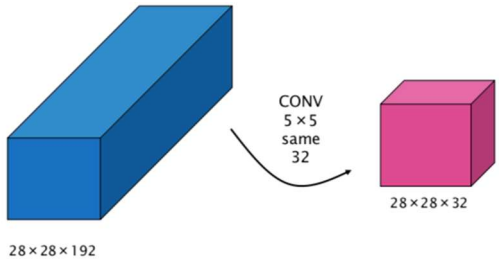
Abbiamo 28x28x192

- Ho tanta potenza di calcolo quindi usando un approccio speculativo **provo più cose per poi scegliere quella che mi da i risultati migliori**
    - 1x1 con 64 filtri
    - 3x3 con 128 filtri
    - 5x5 con 32 filtri
    - Max pooling con stride 1
- I volumi colorati avranno dell'informazione riealaborata

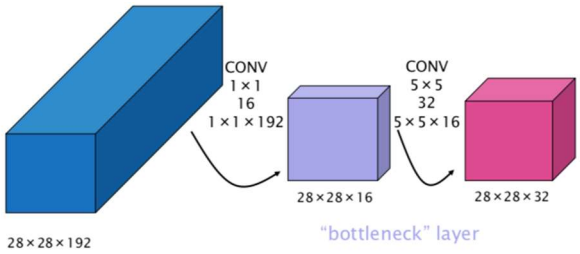


Usiamo convoluzioni same per mantenere le stesse dimensioni.
L'idea dell'inception network è di non dover scegliere che tipo di operazione fare, facendole tutte quante, tanto la rete deciderà di imparare quello che è più utile. C'è però il problema della complessità computazionale.

Prendiamo come esempio la parte di convoluzione con filtro 5x5



Questo richiede circa 120 milioni di moltiplicazioni senza contare le somme.
Per ovviare a questo problema si sfrutta la convoluzione 1x1

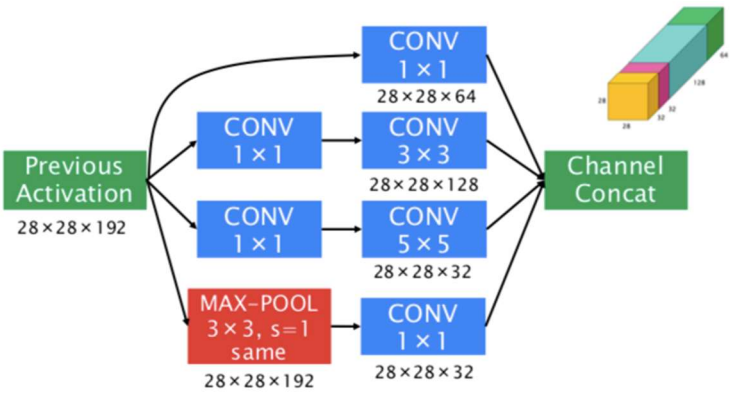


Applico la convoluzione 1x1 ottenendo una profondità di 16 e poi su questo applico la convoluzione iniziale 5x5.

- Con la prima convoluzione si avranno 2,5 milioni di operazioni e con la seconda 10 milioni avendo un totale di circa 12 milioni contro i 120 milioni che sarebbero serviti senza sfruttare la convoluzione 1x1.
- Con questo stratagemma si riduce di un ordine di grandezza il numero di operazioni.
- Non peggiora le prestazioni.
- La notazione bottleneck layer è più riferita alla forma che alla questione prestazionale.

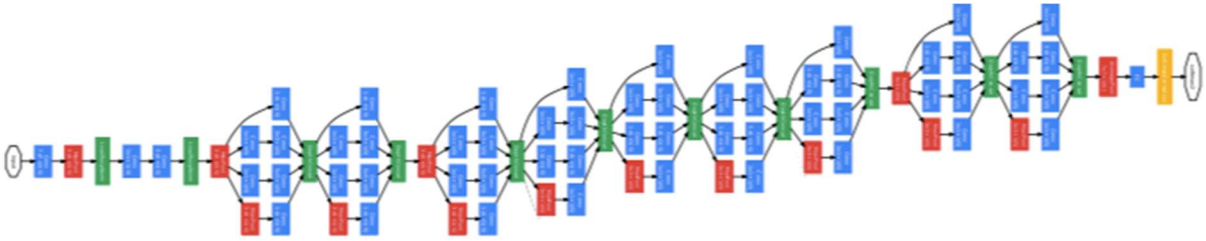
Inception module

Riprendendo l'esempio di prima e applicando la convoluzione 1x1

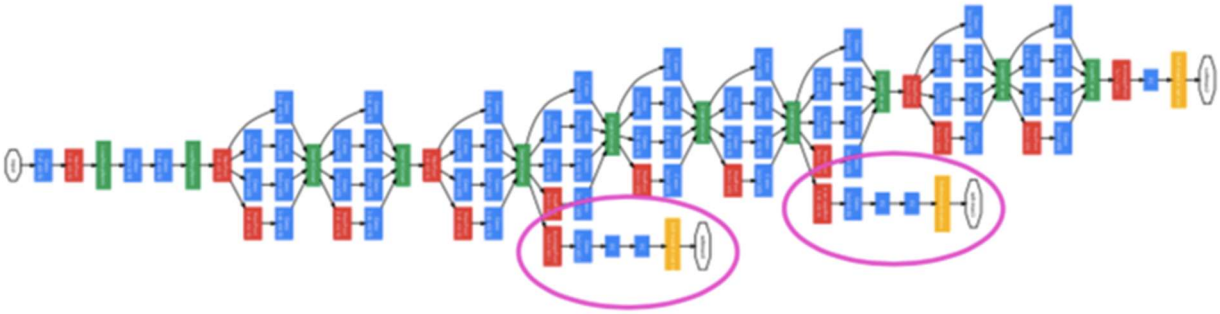


Circa 50 mln di moltiplicazioni ma moltissime informazioni diverse.

Inception networks

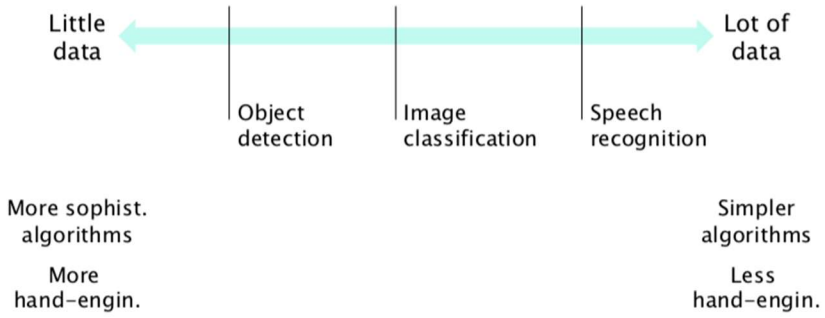


In questo tipo di rete anziché fare la scelta il progettista, sceglie la rete che cosa imparare.
Questa rete ha la caratteristica di essere molto regolare e mi sono tolto il problema di dover scegliere gli hyper-parametri.
Nella rete originale ci sono delle ramificazioni intermedie con un softmax che quindi cercano di predirre in anticipo; hanno un effetto di regolarizzazione.



È stata sviluppata in Google chiamata googLe-Net.
Il motivo per cui la rete funziona molto bene e per cui è stata molto utilizzata è che in una delle challenge ha fatto molto meglio delle altre reti.

Data vs hand-engineering

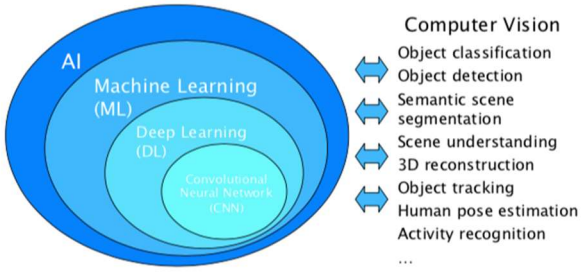


Two sources of knowledge:
Labeled data
Hand engineered features/network architectures/other comp.

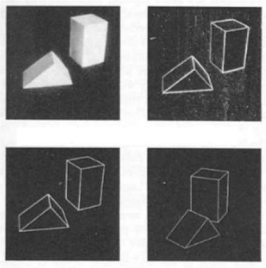
Quando abbiamo a disposizione molti dati i modelli tendono ad essere più semplici non dovendosi inventare troppe cose.
Mentre dove si avevano pochi dati c'è stato bisogno di ricercare e creare dei modelli più sofisticati.
Ci sono due componenti essenziali da cui può imparare il modello:

- dati annotati
- hand-engineered components.

ConvNets e computer vision

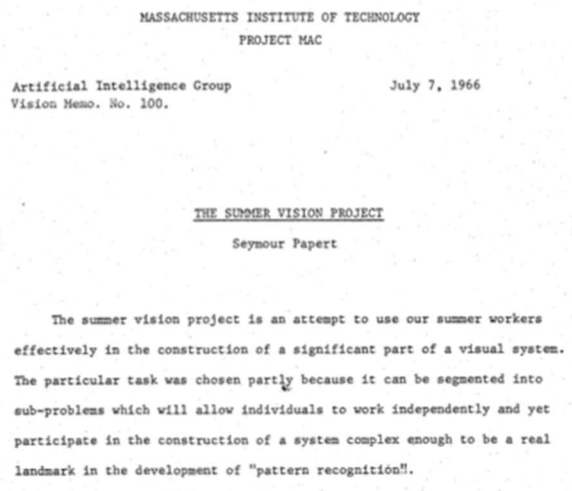


Block word, Larry Roberts 1963



Block word: il mondo ideale può essere diviso in varie forme, il compito della computer visione e riconoscerle.

The Summer Vision Project 1969



MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group — July 7, 1966
Vision Memo. No. 100.

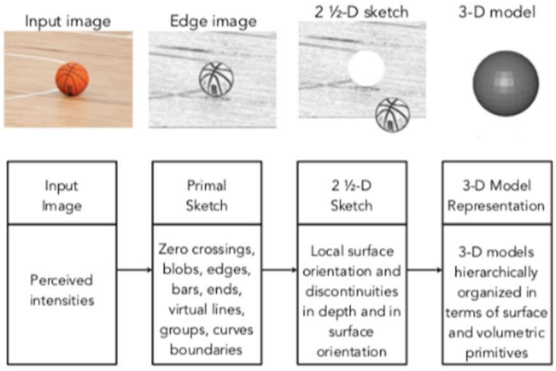
THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

tanti task separati che possono poi portare al riconoscimento nel mondo reale

David Marr



Sistema di visione deve comportarsi come si comporta quello degli animali:
- identificazioni elementi semplici (edges)
- conversione in immagini
- conversione in 3D

David Lowe 1987

Algoritmi per computer vision

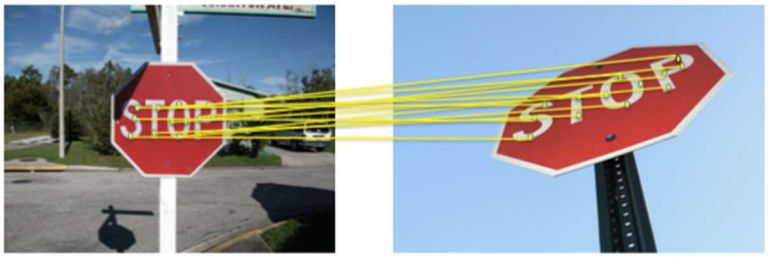
Shi & Malik 1997

Segmentazione: ci sono degli elementi tra di loro collegati, classsifichiamoli in qualche modo.

Viola & Jones 2001

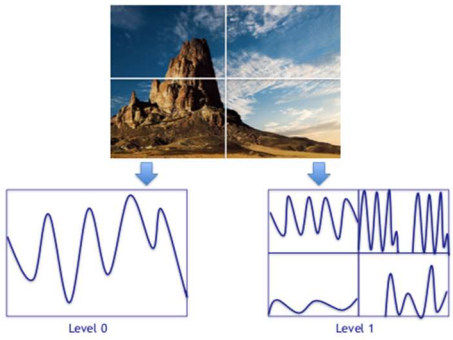
Face detection

David Lowe 1999



Feature che sono punti salienti nell'immagini, scale-variant e rotation-invariant -> per tracciare l'oggetto

Lazebnik, Schmid & Ponce 2006



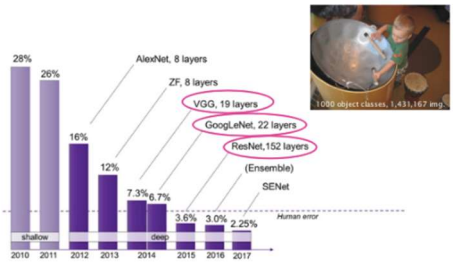
Decomposizione piramidale: studiando la frequenza dell'immagine ne capisco il contenuto

Caltech101/Caltech256

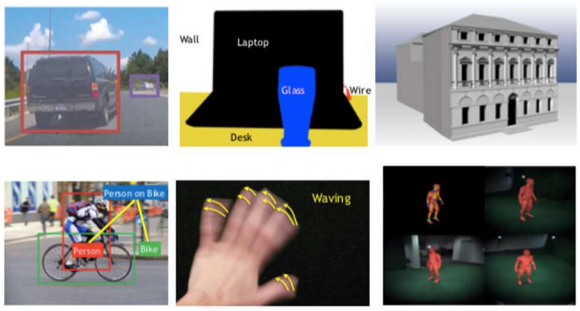
➔ dataset

Pascal VOC Visual Object Classes

ImageNet



Other computer vision tasks

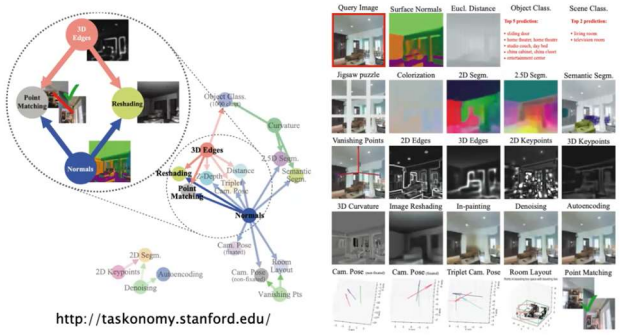


- Object detection
- Segmentazione (con identificazione degli oggetti nella scena)
- Ricostruzione tridimensionale
- Action recognition
- Gesture recognition

PT = 500ms

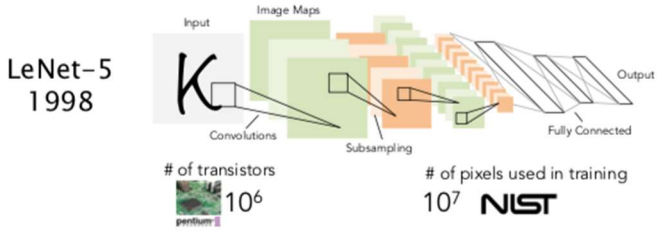
Some kind of game or fight. Two groups of two men? The man on the left is throwing something. Outdoors seemed like because i have an impression of grass and maybe lines on the grass? That would be why I think perhaps a game, rough game though, more like rugby than football because they pairs weren't in pads and helmets, though I did get the impression of similar clothing. maybe some trees? in the background. (Subject: SM)

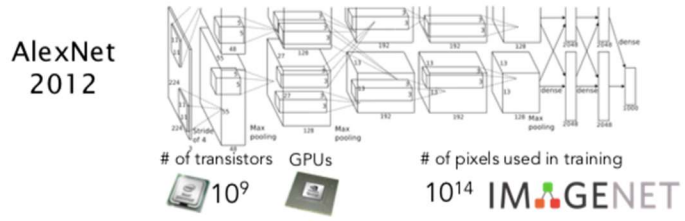
➔ Cosa scriverebbe essere umano vs cosa le intelligenze artificiali disegnerebbero a aprtire dalla descrizione vs cosa dice gpt4 a partire da questa immagine e quanto nel profondo riesce ad andare



➔ Fornisce tassonomia sui task della computer vision
➔ Verifica che sia possibile passare da un task ad un altro sfruttando feature apprese da un altro task

Perché oggi?





È cambiata la capacità di calcolo, sviluppo GPU per paralizzazione, la quantità di dati e la loro grandezza. Nr di pixel usati in training…