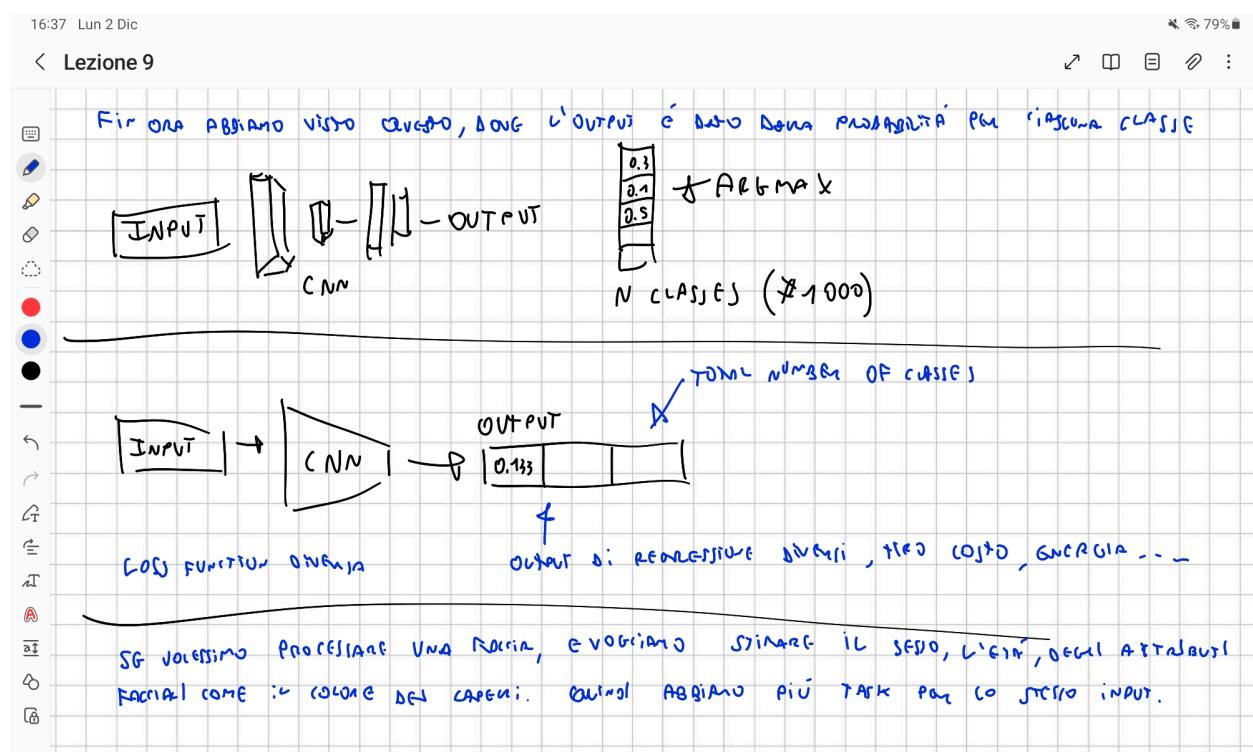


# Lezione 9 - Model compression e video classification -

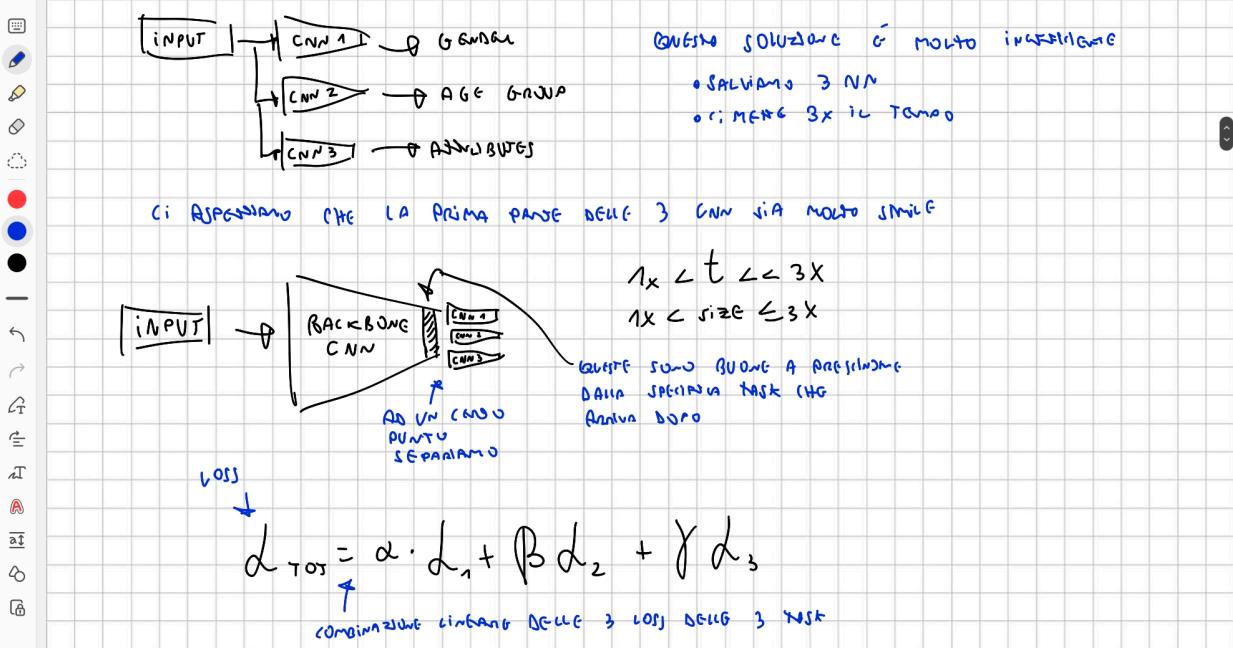
## 02/12/2024

### Lezione 9 - Model compression - 26/11/2024

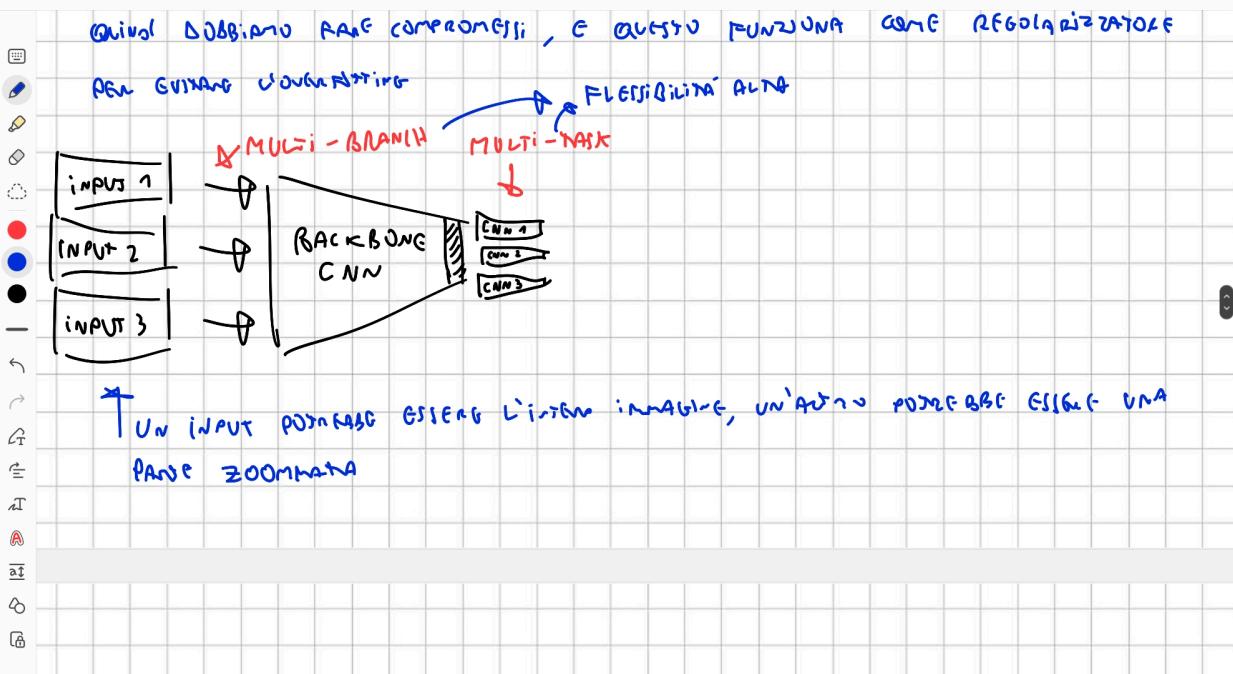
Fino all'esempio delle sinapsi di un bambino.



## &lt; Lezione 9



## &lt; Lezione 9



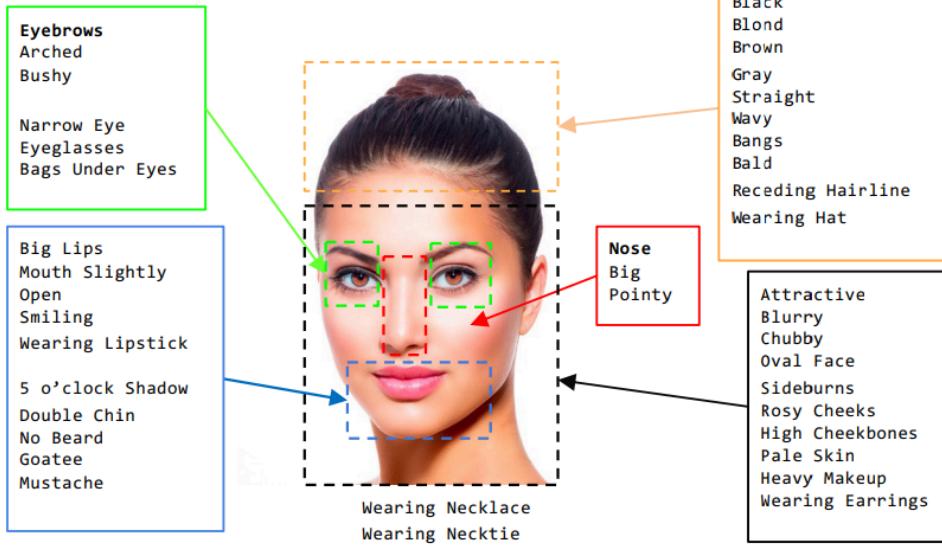
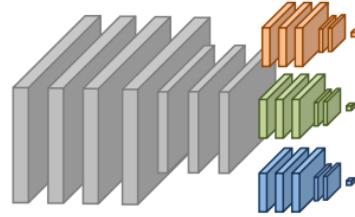
# Facial Attributes

Soft biometrics

Multi-task CNN

40 binary attributes

+130,000 images (CelebA, LFWA)



Female  
25, 32  
Attractive  
Brown Hair  
Heavy Makeup  
High Cheekbones  
Mouth Slightly Open  
No Beard  
Oval Face  
Smiling  
Wavy Hair  
Wearing Lipstick  
Young

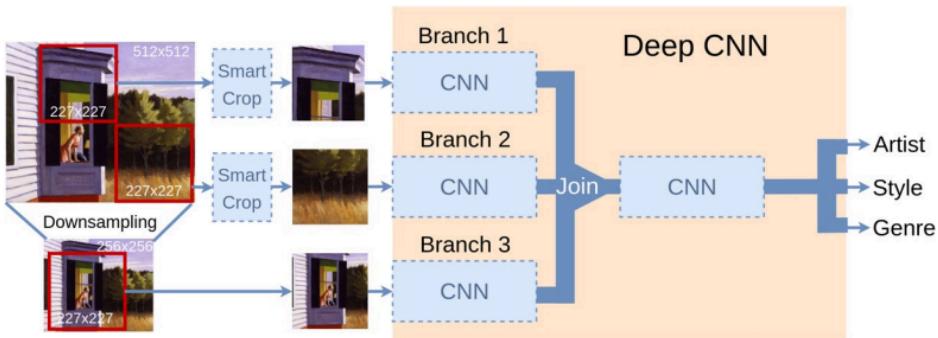
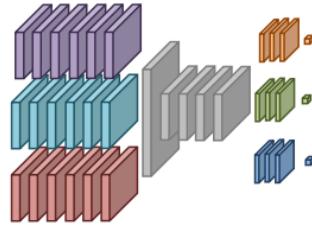


Male  
25, 32  
Attractive  
Bags Under Eyes  
Big Nose  
Black Hair  
Bushy Eyebrows  
Goatee  
High Cheekbones  
Mouth Slightly Open  
Sideburns  
Smiling  
Young

[ivldocker.disco.unimib.it/faceattr/](http://ivldocker.disco.unimib.it/faceattr/)

# Paintings

Painting Similarity  
Multibranch Deep CNN  
1508 artists, 125 styles, 41 genres  
+100,000 images (WikiPaintings-IVL)



Artist	Style	Genre
Paul Gauguin	0.99931	
Paul Serusier	0.00068	
Grégoire Michonze	0.00000	
Christopher Wood	0.00000	
Yiannis Tsarouchis	0.00000	

Artist	Style	Genre
Cloisonnism	0.99842	
Synthetism	0.00100	
Naïve Art (Primitivism)	0.00041	
Post-Impressionism	0.00014	
Social Realism	0.00001	

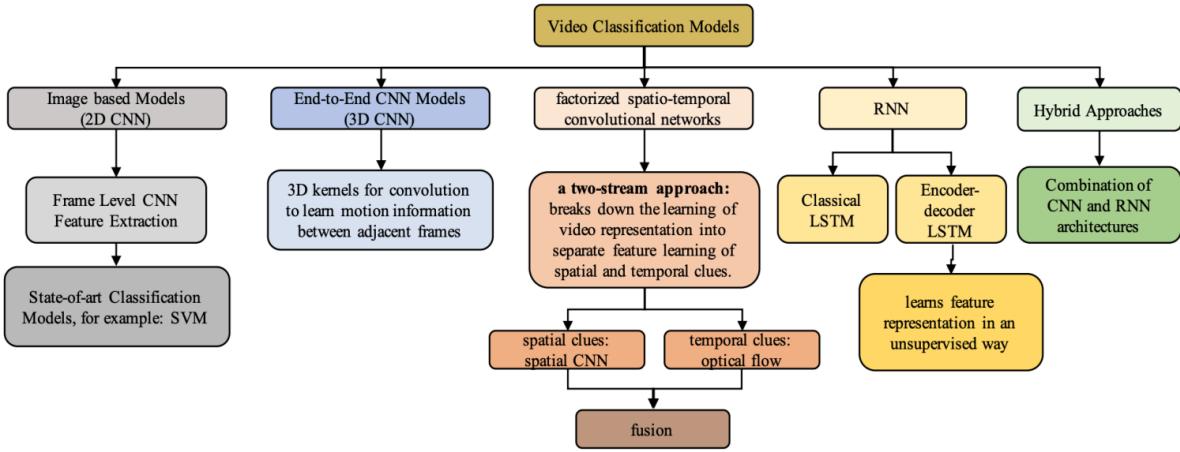
Artist	Style	Genre
genre painting	0.86280	
portrait	0.09878	
self-portrait	0.03626	
mythological painting	0.00099	
symbolic painting	0.00050	

ivldocker.disco.unimib.it/paintings/

## Video processing with Recurrent Neural Networks

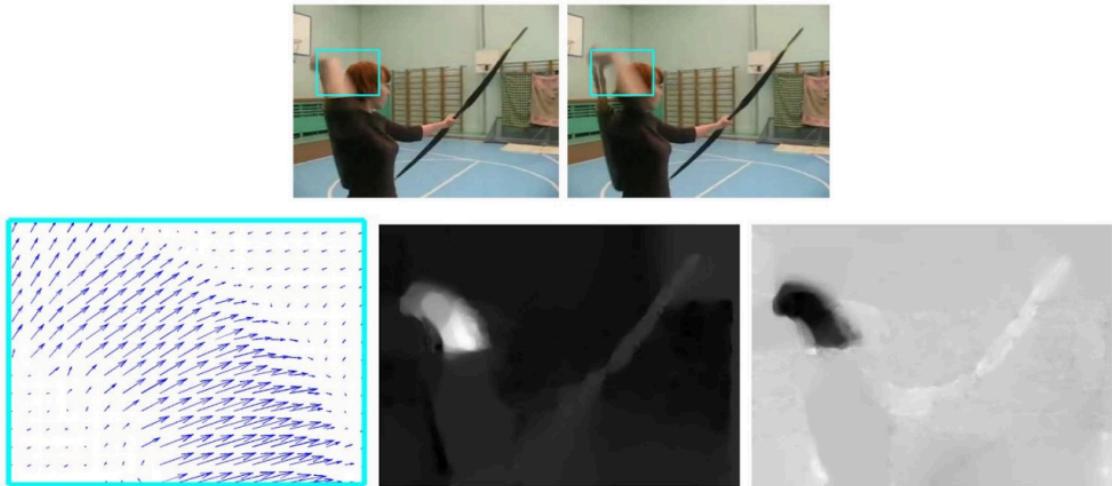
Quando vogliamo classificare il contenuto di un video, dobbiamo capire che abbiamo una dimensione addizionale che dobbiamo considerare: il **tempo**. Ad ogni time step abbiamo un'immagine bidimensionale.

Oggi vedremo un sommario dei modelli per la classificazione dei video:

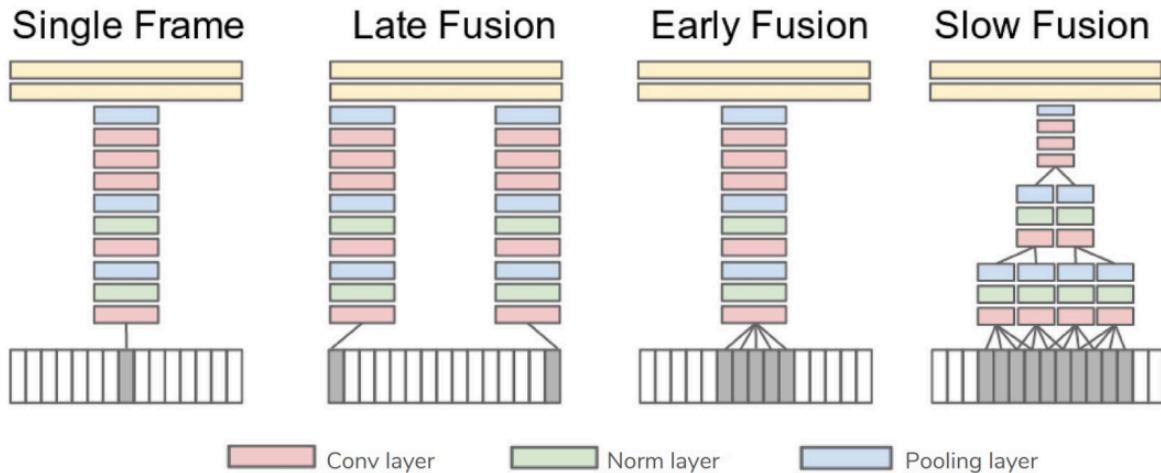


Innanzitutto, perchè non usiamo un modello normale (colonna grigia)? Questo è quello che si faceva prima del deep learning, si trovavano delle features locali che seguivano il displacement di un oggetto da un frame al prossimo. Quindi facciamo un encoding di come gli oggetti si muovono.

Poi abbiamo approcci di traiettoria, ovvero accumulando quest'informazione su multipli frames, troviamo un'informazione di questo tipo:



Anche in questo approccio, abbiamo diversi modi in cui fondere le informazioni dei diversi frame.



- **Single Frame:**

- Analizza solo un fotogramma alla volta senza integrare informazioni temporali.
- Il modello è semplice ma ignora la dinamica temporale della sequenza.

- **Late Fusion:**

- Ogni fotogramma viene processato separatamente attraverso una serie di strati convoluzionali.
- L'informazione temporale viene combinata solo alla fine, usando un livello di fusione.
- Approccio semplice che cattura le caratteristiche per ogni frame prima di combinarle.

- **Early Fusion:**

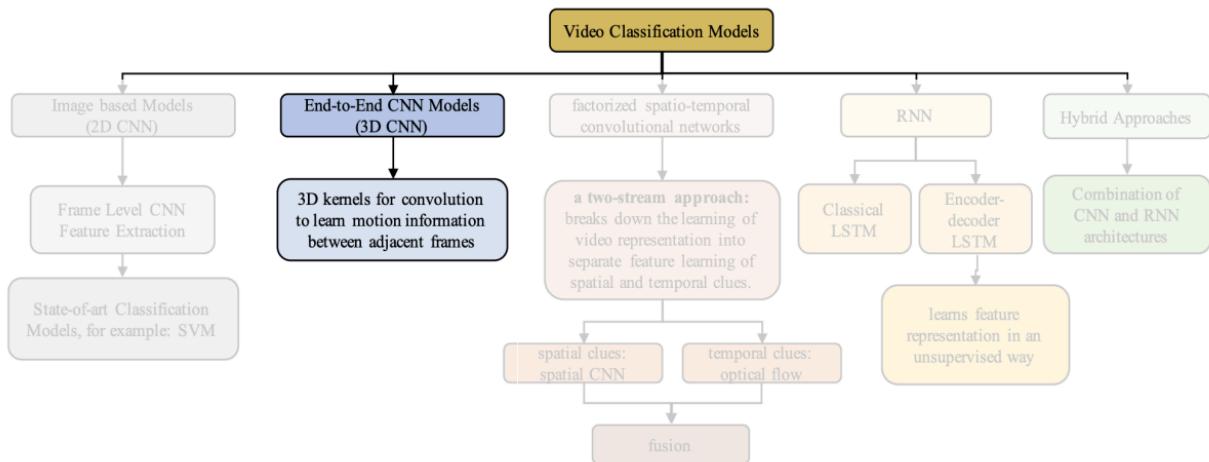
- Combina l'informazione temporale nei primi strati del modello, trattando i fotogrammi come un unico input volumetrico.
- Permette al modello di considerare la dinamica temporale sin dalle prime fasi di elaborazione.

- **Slow Fusion:**

- Integra l'informazione temporale in modo graduale lungo i vari strati del modello.

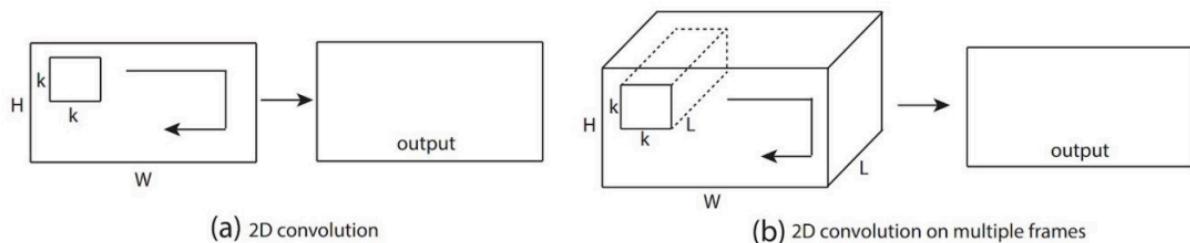
- Fornisce un equilibrio tra analisi temporale e spaziale, migliorando la rappresentazione complessiva delle sequenze.

## End-to-end CNN models: 3D CNNs



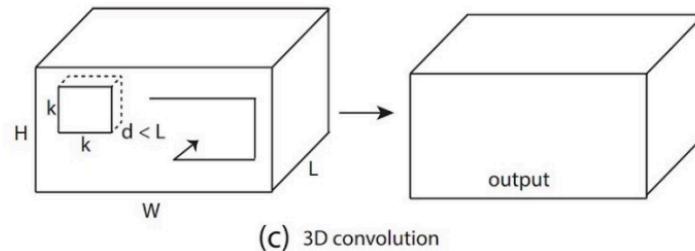
Un altro modello sono le CNN tridimensionali.

In una convoluzione tipica abbiamo che il filtro fa uno scan dell'input per darci l'output. Se abbiamo un input che è composto da più frames, la convoluzione tradizionale collassa l'informazione temporale. Ciascun filtro lavorerà su tutti i livelli contemporaneamente, schiacciando la dimensione temporale.



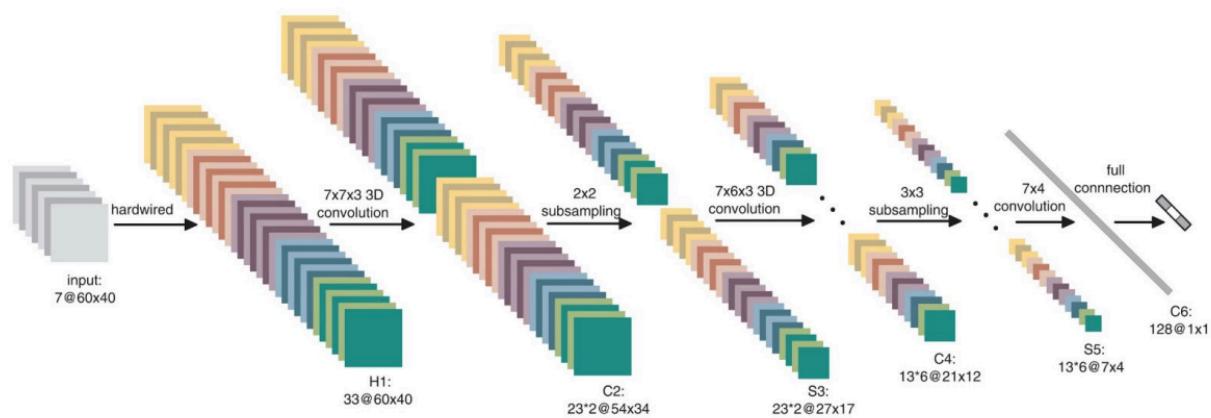
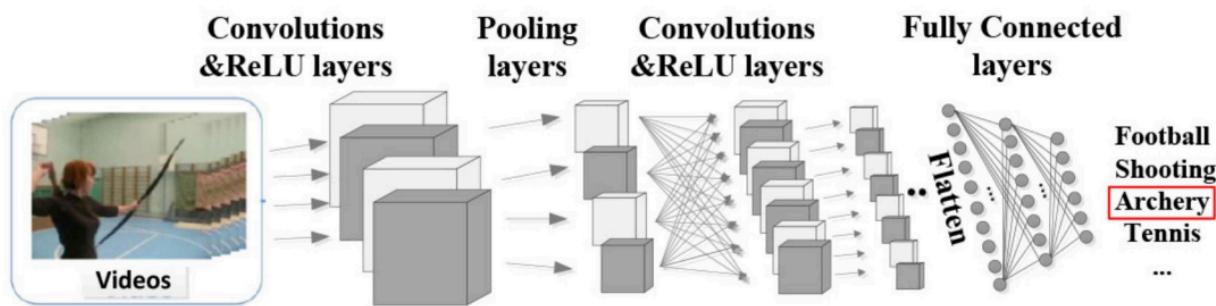
Ora la differenza è che ora abbiamo che il filter depth e l'input depth non sono più per forza identiche, quindi questa non è più un'operazione full-in-depth.

- A 3D filter can thus move in all the three dimensions
- The output is thus a 3D data structure for each filter (instead of just a 2D plane)



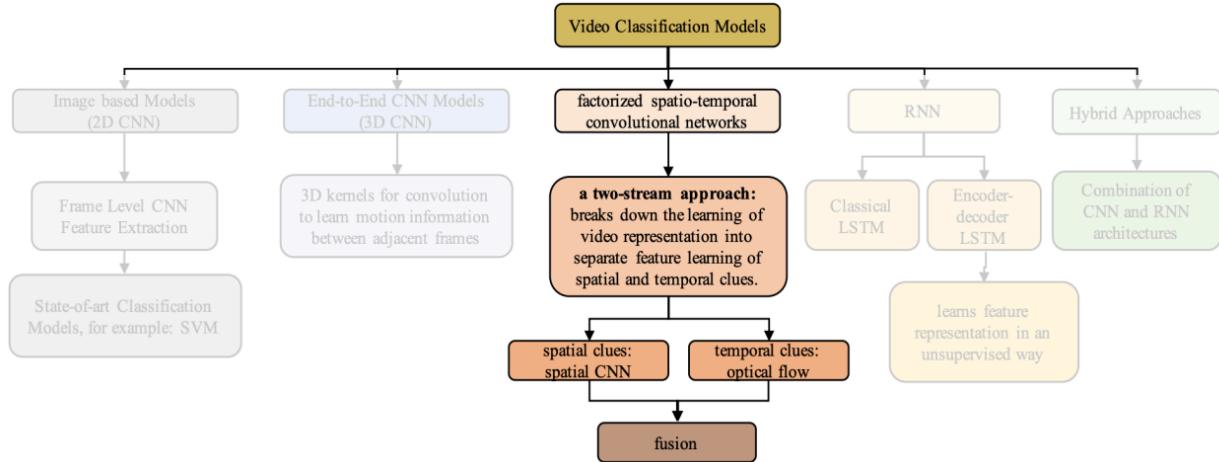
Quindi avremo un volume in output, perchè ad ogni time step abbiamo un depth channel.

Per esempio questa è una rappresentazione di un NN:



Abbiamo una nuova dimensione sulla quale il filtro può fare lo scan.

## Factorized spatio-temporal convolutional NN

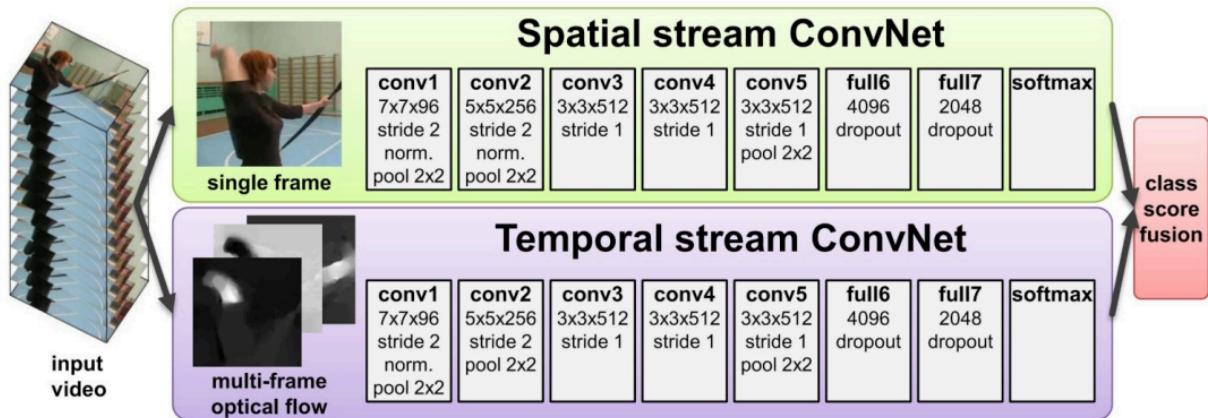


Qui abbiamo lo sfruttamento del fatto che il video è una combinazione dell'apparenza e del movimento.

$$\text{VIDEO} = \text{Appearance} + \text{Motion}$$

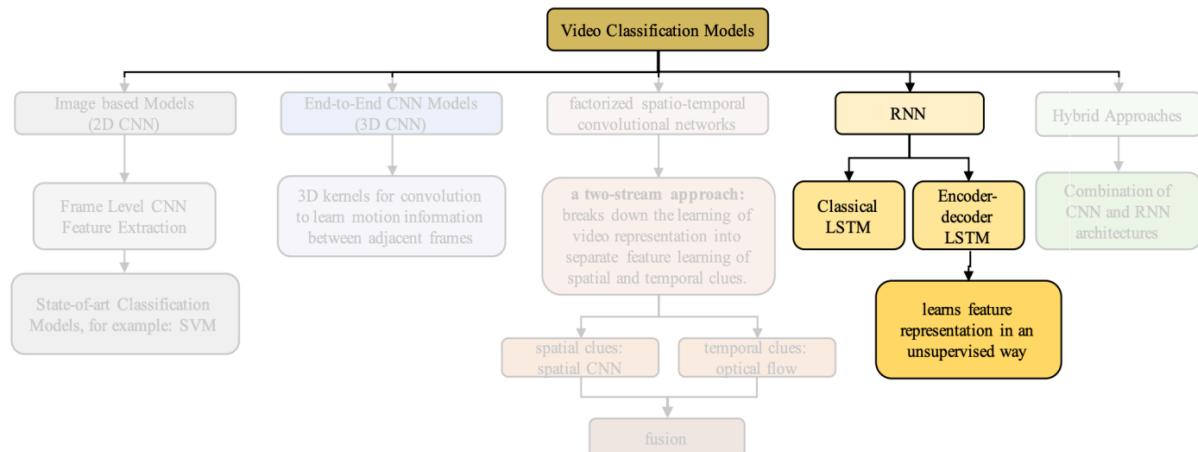
- They contain complementary information:
  - Single frames: static appearance
  - Multi-frame: e.g. optical flow: pixel displacement as motion information

In questo esempio selezioniamo un frame che è processato da una CNN (per l'informazione spaziale), e nell'optical flow descriviamo come gli oggetti si stanno muovendo (per l'informazione temporale).



Alla fine li fondiamo, quindi è un'approccio late-fusion.

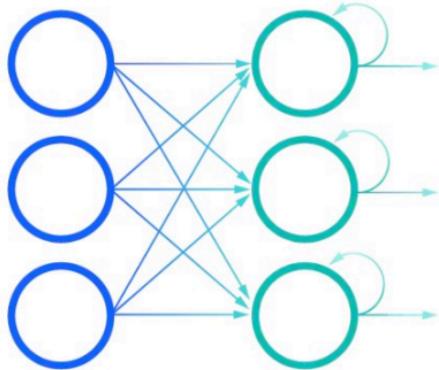
## RNN



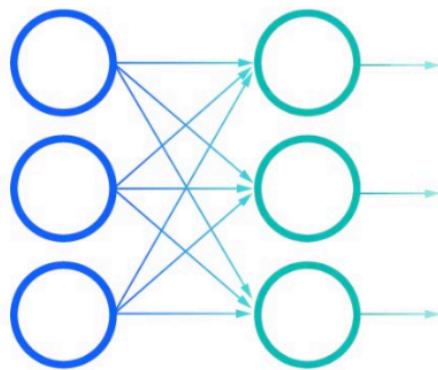
Il terzo tipo di approccio usa le recurrent neural networks, che sono una famiglia di NN usate per processare dati sequenziali.

Le RNN danno la possibilità di processare input con una lunghezza diversa.

RNN



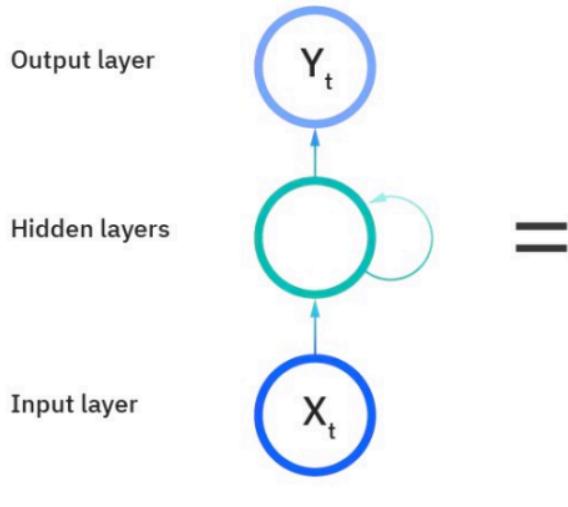
Classical FNN



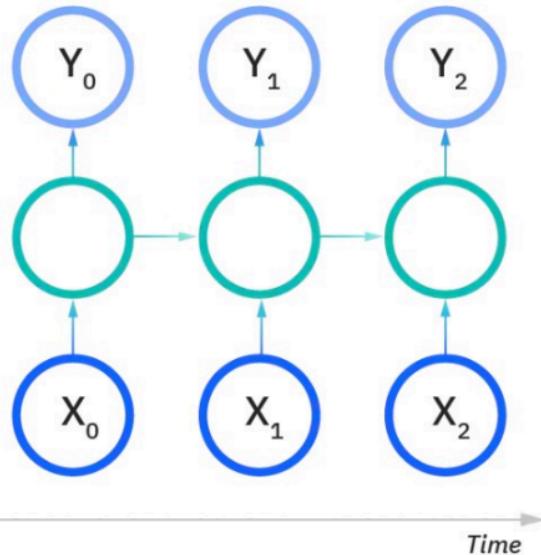
Sembra che un neurone sia connesso con se stesso.

Facendo l'unfolding nel tempo, producendo la vista unrolled, vediamo che per la nostra predizione al tempo 1, abbiamo dell'informazione che arriva dal processing del time step precedente. Quindi stiamo mantenendo dell'informazione (memoria) da usare nei prossimi time steps.

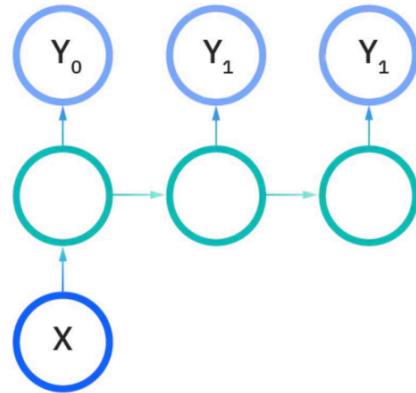
Rolled RNN



Unrolled RNN



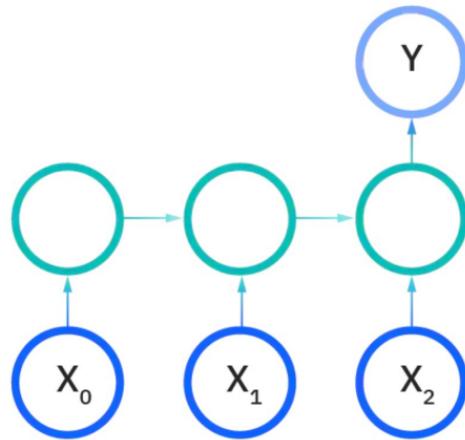
Architettura one-to-many:



Abbiamo un input, e vogliamo produrre più outputs. Per esempio abbiamo un'immagine in input, e vogliamo dare una descrizione testuale dell'immagine.

---

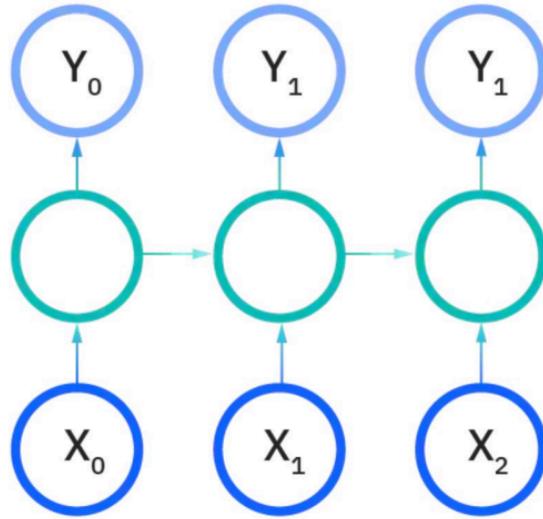
Architettura many-to-one:



Un use case può essere la classificazione del video, o del topic del testo input.

---

Architettura many-to-many:



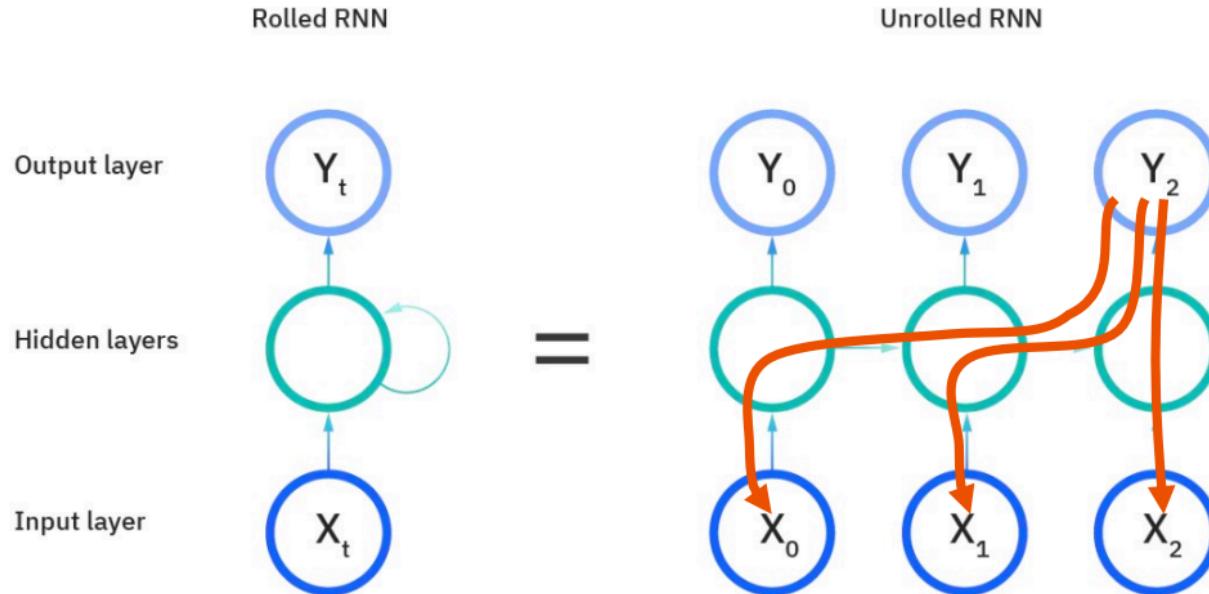
Questo può essere video classification ma frame per frame, perchè abbiamo più azioni che vogliamo riconoscere.

## Training

Per allenare il network dobbiamo fare la backpropagation though time.

L'output  $Y_2$  dipende dall'input  $X_2$  ma anche dagli input precedenti  $X_1$  e  $X_0$ , quindi il gradiente deve backpropagare anche in quelle 2 direzioni.

Questo rende il RNN molto profondo.

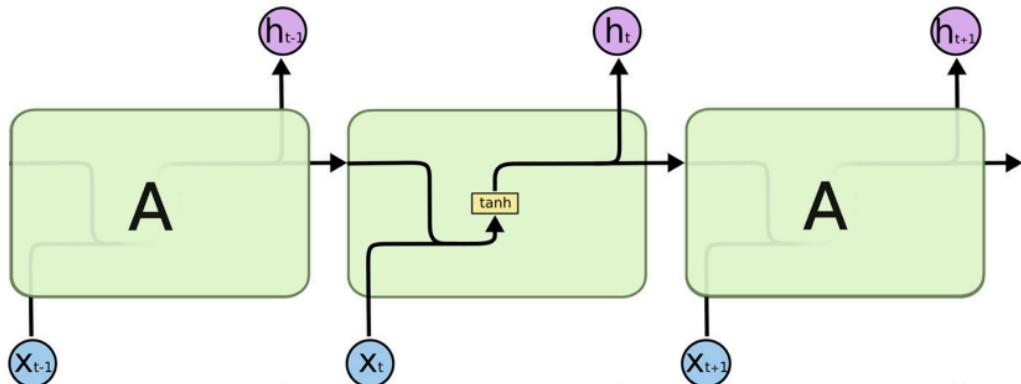


Questo rende peggiore il problema del vanishing gradient.

è stata creata l'architettura delle **gated RNNs** per risolvere questo problema.

## Gated RNNs - LSTM

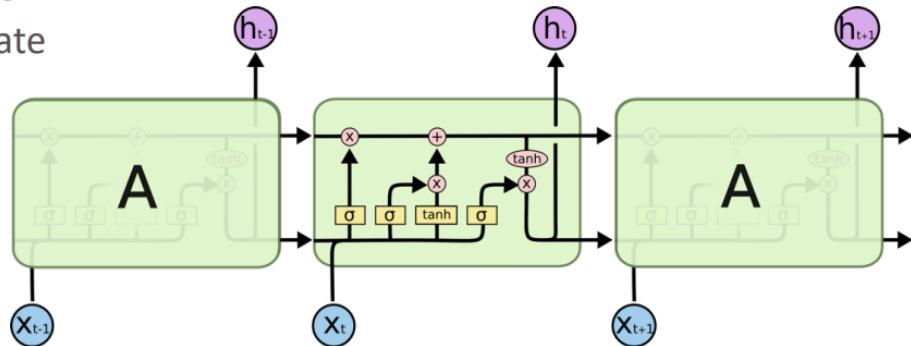
Questa è una normale RNN, che prende concatena l'input con il dato del time step precedente, li fa passare per un tanh...



Quindi quando facciamo backpropagation dobbiamo passare per un fully connected layer (tanh). Quindi il gradiente diminuisce più andiamo indietro nel tempo.

Nell'architettura LSTM (Long Short-Term Memory) sostituiamo l'hidden unit con questa:

- Forget gate
- Input gate
- Output gate



Abbiamo un'internal state variable che è quella nella riga superiore, e 3 gates diversi che modificano l'informazione della cell state.

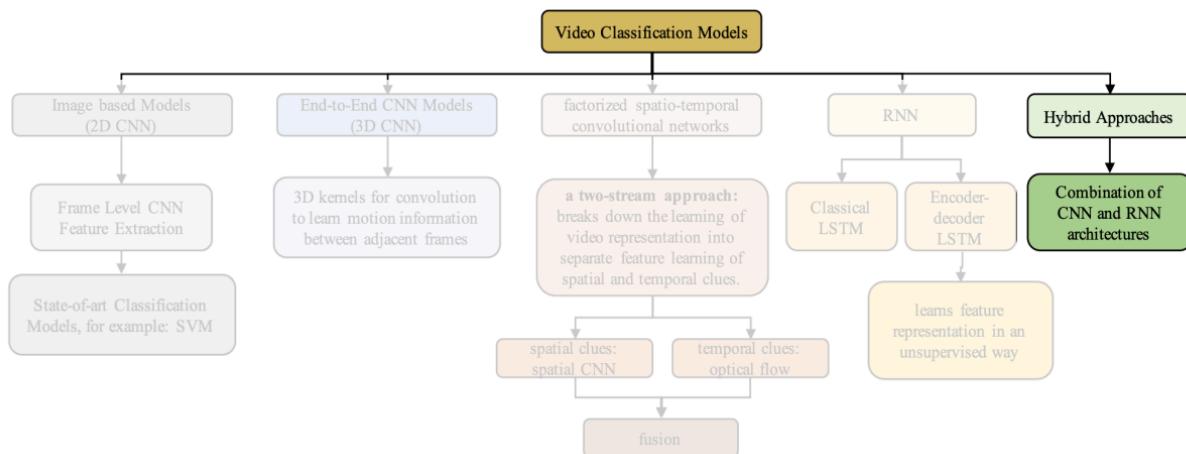
Il forget gate elimina l'informazione non importante.

L'input gate controlla quale informazione dell'input deve essere aggiunta al cell state.

L'output gate computa la nuova informazione da passare al prossimo time step

Quindi ora per backpropagare possiamo passare da sopra, dove il gradiente può passare senza modifiche.

## Approcci ibridi



Gli approcci ibridi combinano le CNN e le RNN.

Questa è una multi-stream network che fa diverse operazioni in modi diversi sullo stesso video. Da una parte abbiamo operazioni su singoli frames fatti da CNN tradizionali. Le features in output da questi CNN vengono usati come input di una LSTM, che è la RNN del lato spaziale del video.

Poi partendo dall'optical flow otteniamo informazione che riguarda solamente il movimento. Questa va in una seconda CNN dove l'attivazione è presa come input per il secondo motion LSTM che ci dà altra informazione.

Nei video di solito c'è anche l'informazione dell'audio, quindi si prende l'audio (che è un'informazione monodimensionale) e lo si trasforma in uno spettrogramma per

renderlo 2D, così abbiamo informazioni sulla frequenza ad ogni time step. Quest'informazione può essere processata da una terza CNN dedicata all'audio. Quindi abbiamo un'informazione spaziale, una di movimento e una di audio, che sono messe insieme per raggiungere la nostra classificazione finale.

