

Lezione 14 - Self-Supervised Learning - 17/12/2024

Transformers can process sequences of arbitrary length

Congratulations!

True 

False

Transformers process each input independently

True

False

The only processing module in a transformer layer is self attention



Congratulations!



True

False



Self attention, being a composition of two linear transformations is linear

True

False

The elements that are needed for the computation of self attention are:



Congratulations!



Outputs

Inputs



Values



Variables

Queries



Questions

Keys



Chains

A transformer head is completely defined by 3 weight matrices and 3 biases



Congratulations!



True



False

To obtain the best performance usually transformers use one single head



Congratulations!



True



False

Self-Supervised Learning

Sostanzialmente, abbiamo visto che per molte task i deep NN superano i metodi precedenti di molto, e ora vengono usati per risolvere quasi tutti i problemi.

In particolare, questi modelli sono allenati da dataset molto grandi come imagenet, il modo più comune è quello di iniziare da pre-trained models e poi fare fine tuning per la task specifica.

Quest'idea è per 2 motivi: i pesi sono già in una buona configurazione perché sono imparati su un dataset molto grande, quindi è una **hot start**, anche se la soluzione non è ottima per il problema specifico. D'altro canto, imparando da molti dati, la rete ha già imparato una gerarchia dei dati.

Quindi questo aiuta a non overfittare. Questo è vero specialmente se il nostro dataset è molto piccolo.

La prestazione delle CNN profonde (ConvNets) dipende fortemente dalla loro capacità e dalla quantità di dati di addestramento.

La combinazione di architetture sofisticate e dataset grandi, ha permesso di migliorare i risultati di praticamente qualsiasi task.

Il problema è che la creazione di dataset grandi è molto costoso a livello di tempo.

Per esempio ImageNet ne contiene 1.3 image su 1000 classi, che sono labellati manualmente da persone. Si può immaginare come costoso può essere l'annotazione di un dataset video che per esempio ha le labels delle azioni che vengono fatte. Mettere i label a tutte queste informazioni è molto costoso, va fatto manualmente.

Per provare ad evitare quest'annotazione dei dati costosa, spesso si usano dei metodi **self-supervised**.

Una soluzione popolare è quella di avere una “**pretext-task**” che la rete deve risolvere. Nel frattempo che la rete sta imparando per questa task, sta imparando anche informazioni utili per la nostra task specifica.

Le **pretext-task** anno 2 proprietà:

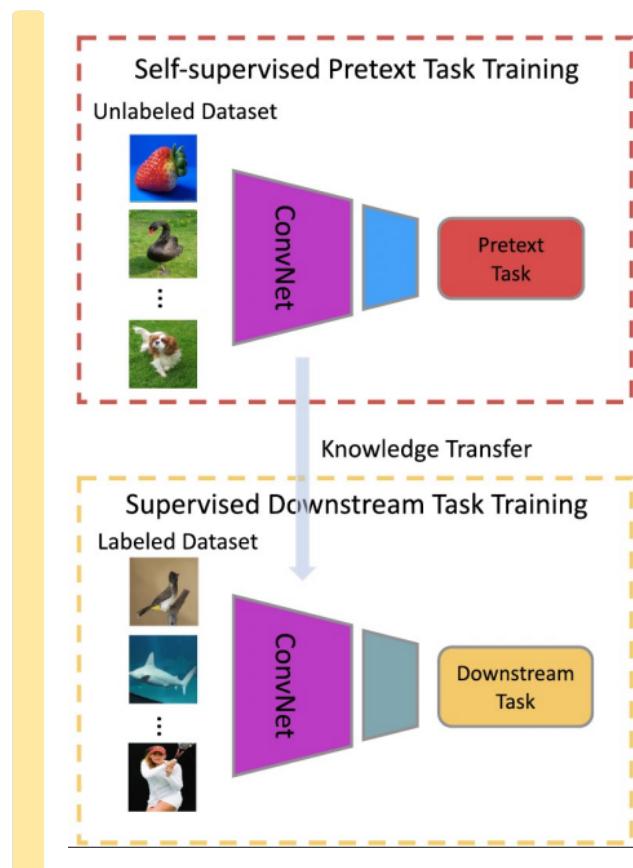
- Mentre vengono risolti questi problemi, dobbiamo imparare
- I labels per queste task devono essere generati dai dati stessi, quindi ci deve essere una proprietà interna dei dati che non deve essere labellata da una persona.

- (1) visual features of images or videos need to be captured by CNNs to solve the pretext tasks,
- (2) the supervisory signal is generated from the data itself (self-supervision) by leveraging its structure.

La CNN parte da un unlabelled dataset.

Poi si applica un knowledge transfer verso la nostra task.

Quindi trainiamo la rete in una dataset grande di cui abbiamo le labels. Quando è trainato, possiamo fine-tunarlo con la nostra task che ha un dataset più piccolo labellato.



Terminologia

Le **human-annotated labels** sono quelle create da persone.

La **pretext task** è una task che la rete deve risolvere, da cui la rete deve imparare delle **features visuali**. Ce ne sono di diversi tipi, possono essere predittive,

generative, contrasting o un mix. La cosa importante è che la label arriva da un qualcosa già interno al dato.

Le pesudo labels sono le labels estratte dai dati.

La downstream task è la task che vogliamo risolvere, è il problema reale, per il quale per esempio non abbiamo abbastanza dati per trainare il model oppure non abbiamo abbastanza dati labellati.

Il supervised learning si riferisce a tutti i metodi che usano **human-annotated labels**.

Il semi-supervised learning è un tipo di metodo dove abbiamo un piccolo numero di dati labellati, insieme ad un grande numero di dati non labellati.

Il weekly-supervised learning è un metodo che utilizza labels "coarse-grained" (meno dettagliate, generiche) oppure **inaccurate**. Il costo per ottenere labels è generalmente molto più economico rispetto alle etichette a "grana fine" per metodi supervisionati.

Con inaccurate si intende che per esempio possiamo usare un metodo che fa labels molto velocemente ma magari non è super accurato.

L'unsupervised learning è dove impariamo senza usare nessuna label human-annotated, non abbiamo proprio le labels

Mentre nel **self supervised learning** estraiamo la label dai dati, e li usiamo per trainare, sfruttando la struttura stessa dei dati.

Vantaggi

Visto che non abbiamo human-annotation, uno dei vantaggi principali è quello che questo metodo può essere usato su dataset di qualsiasi dimensione, perché abbiamo bisogno solo i dati, non le labels.

I metodi self-supervised raggiungono o anche superano la performance di metodi tradizionali.

Viene usato per trainare i **foundation models**, ovvero quei modelli di partenza con features generali e robuste che possiamo usare per molte task diverse.

- Many pretext tasks have been designed and applied for self-supervised learning such as:
 - foreground object segmentation,
 - image inpainting,
 - clustering,
 - image colorization,
 - temporal order verification,
 - visual audio correspondence verification, etc.

Molti pretext-tasks sono stati progettati e applicati per l'apprendimento auto-supervisionato, come:

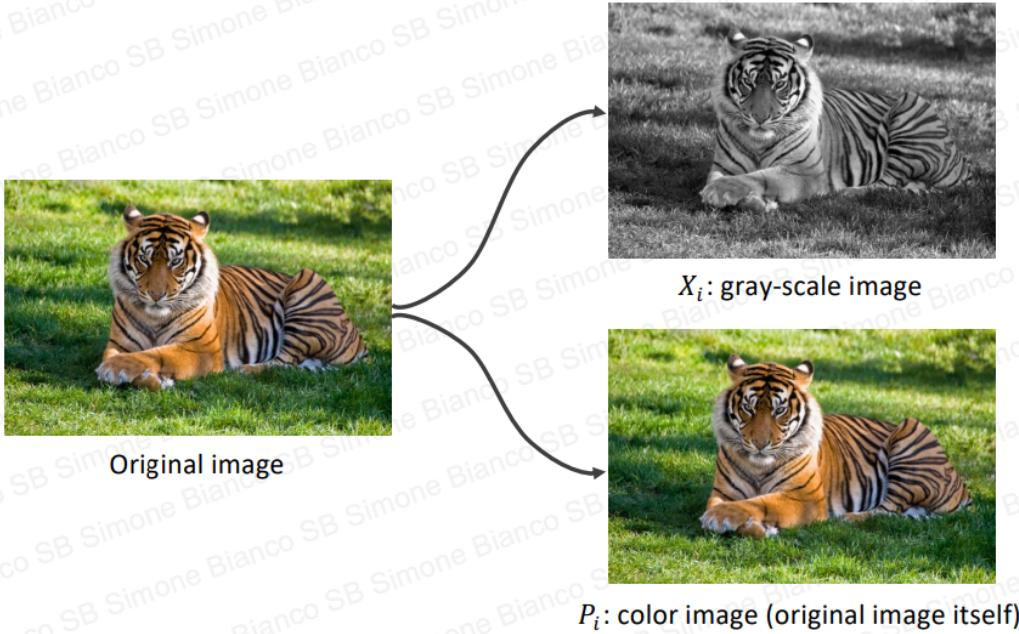
- foreground object segmentation,
- image inpainting (è quello dove manca un pezzo dell'immagine che viene ricostruito dal modello),
- clustering,
- colorazione delle immagini,
- verifica dell'ordine temporale,
- verifica della corrispondenza audio-visiva (associa l'audio al video), ecc.

Vediamo un esempio su colorization, dove partendo da un'immagine in scala di grigi, la coloriamo.

Per generare delle immagini realistiche, la rete deve imparare la struttura e il contesto delle immagini, deve imparare che questa è una tigre, che ha quei colori... che quella è erba ed è verde...

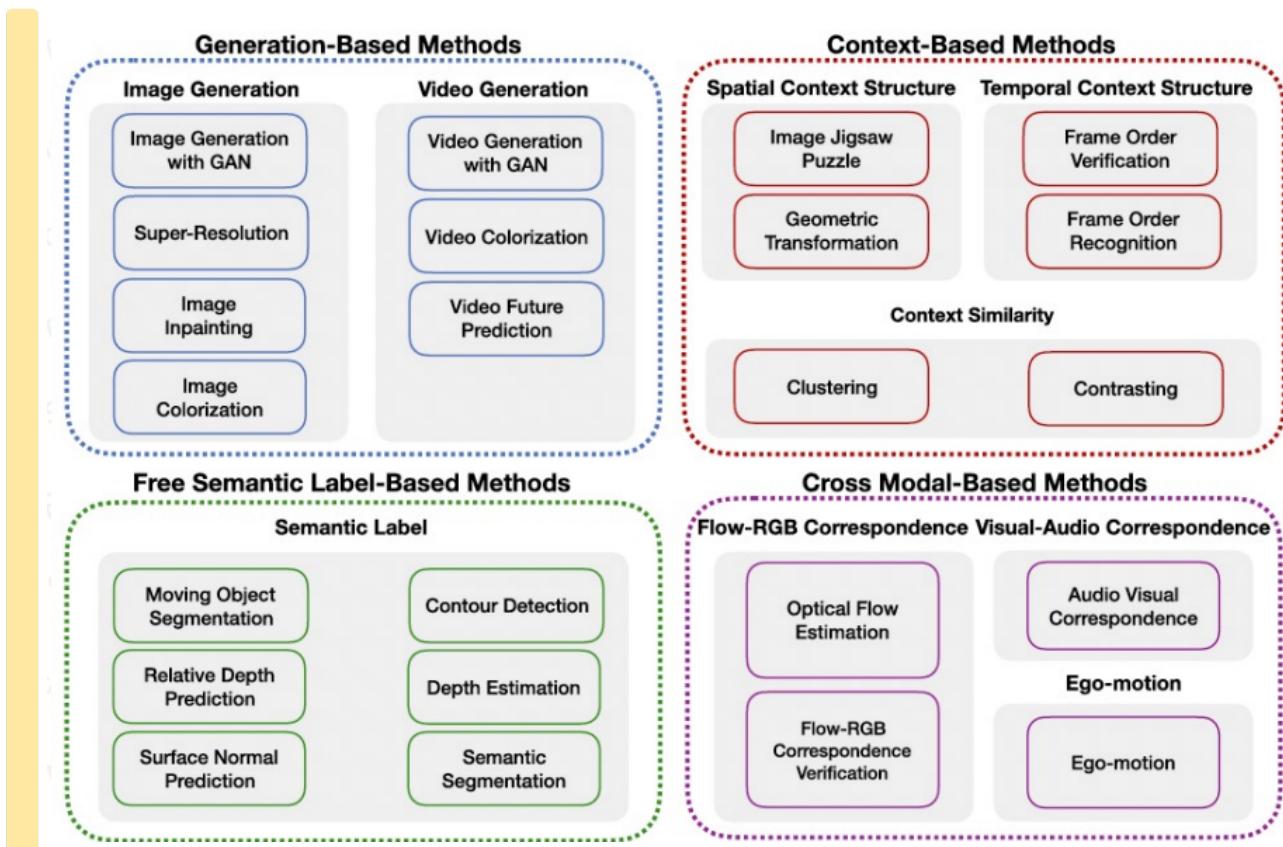


Quindi possiamo avere quante immagini vogliamo, possiamo anche generarle sul momento, a basso costo.



Commonly used pretext tasks

Ciascuna pretext-task sfrutta qualcosa all'interno dei dati. Abbiamo 4 tipi:



Generation-based pretext tasks

Questo tipo di metodi impara features visuali utilizzando pretext tasks che riguardano la generazione di immagini o video.

- **Image generation:** le features sono imparate attraverso il processo della task di generazione dell'immagine. Questo tipo di metodi include la colorazione delle immagini, la super resolution, l'image inpainting, l'image generation con GANs.
- **Video generation:** le features sono imparate attraverso il processo delle task di generazione video. Questo tipo di metodi include la generazione di video con GANs e il video prediction.

Quindi le GANs possono essere usate sia per generare che come pretext-task.

Super-resolution è dove riduciamo la grandezza di un'immagine, e usiamo quella originale come target, aumentando la risoluzione.

Context-based pretext tasks

Nei metodi **context-based pretext** abbiamo che il pretext-task principalmente sfruttano il features di contesto come:

- **Context similarity:** i compiti di pretesto sono progettati in base alla **somiglianza contestuale** tra le porzioni di immagine. Questo tipo di metodi include metodi basati su **clustering di immagini** e metodi constraint-based sui grafi .
- **Spatial context structure:** le pretext class usate per addestrare le CNN si basano sulle **relazioni spaziali** tra le porzioni di immagine. Questo tipo di metodi include i **puzzle jigsaw**, la previsione del contesto e riconoscimento delle trasformazioni geometriche, ecc.
- **Temporal context structure:** l'ordine temporale dei video è usato per la supervisione. La CNN è addestrata a verificare se la sequenza dei frame in ingresso è nell'ordine corretto o a **riconoscere l'ordine della sequenza dei frame**.

Free semantic label-based pretext class

Queste task trainano la rete con **labels** che sono **generate automaticamente**, da algoritmi oppure anche da game engines. Sono labels che possono essere **estratte dai dati senza dover trainare nulla**.

Includono metodi come moving object segmentation, contour detection, relative depth prediction...

Cross modal-based pretext tasks

In questo tipo di metodi abbiamo multiple informazioni multi-modali. Per esempio abbiamo su un canale l'informazione audio e in un altro quella video. Vogliamo separarli e per esempio trovare il sample audio che corrisponde al video, oppure RGB-Flow Correspondence Verification, Contrastring e egomotion.

Commonly used downstream tasks for evaluation

Se abbiamo pre-trainato 2 modelli in 2 modi diversi, come possiamo sapere qual è quello migliore?

I pesi appresi tramite SSL vengono utilizzati come modelli pre-allenati e successivamente fine-tunati su attività **downstream** come la classificazione delle

immagini, la segmentazione semantica, il rilevamento di oggetti e il riconoscimento delle azioni, ecc.

Le prestazioni attività downstream dimostrano la generalizzabilità delle caratteristiche apprese. Se le CNN allenate in scenari SSL possono apprendere caratteristiche generali, allora i modelli pre-allenati possono essere utilizzati come un buon punto di partenza per altre attività di visione che richiedono la cattura di caratteristiche simili da immagini o video.

La classificazione delle immagini, la segmentazione semantica e il rilevamento degli oggetti sono solitamente utilizzati come attività per valutare la generalizzabilità delle caratteristiche delle immagini apprese tramite metodi SSL, mentre il riconoscimento delle azioni umane nei video viene utilizzato per valutare la qualità delle caratteristiche video ottenute dai metodi SSL.

Generation-based Image Feature Learning

Abbiamo un'immagine, e vogliamo generarne un'altra per esempio con del contenuto che mancava oppure una versione colorata...

Di solito su questa task il ground truth è l'immagine originale, da cui abbiamo creato una variante per la nostra pretext class.

Un auto encoder è un image-based generator method che comprime l'immagine in un latent space e poi riesce a ricostruire l'immagine.

Image generation with GAN

- Generative Adversarial Network (GAN) is a type of deep generative model that was proposed by Goodfellow et al.
- A GAN model generally consists of two kinds of networks:
 - 1) a generator, which is to generate images from latent vectors,
 - 2) and a discriminator, which is to distinguish whether the input image is generated by the generator.

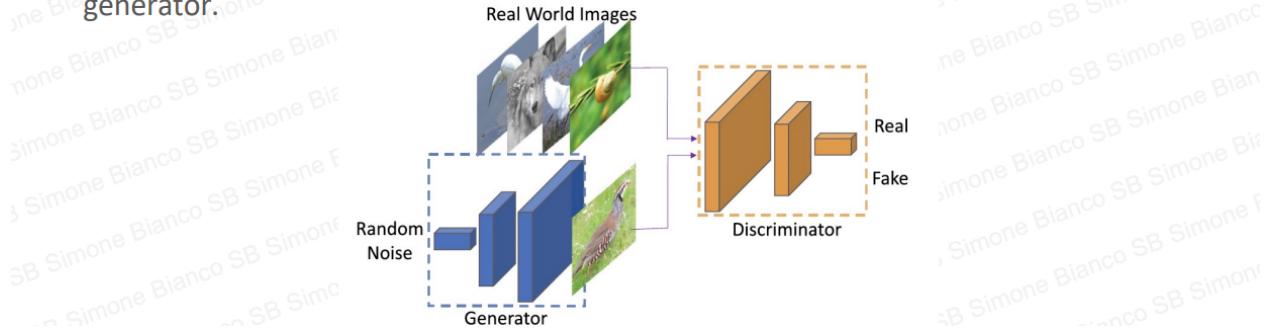
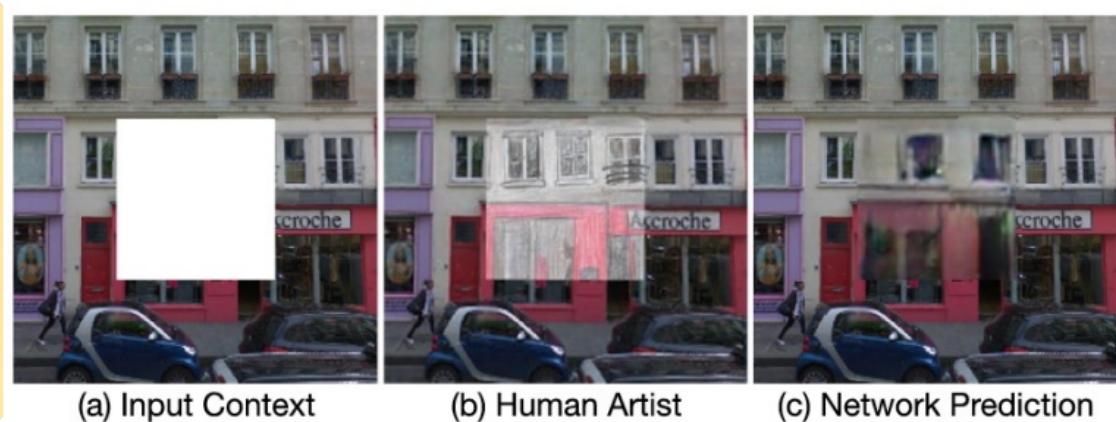


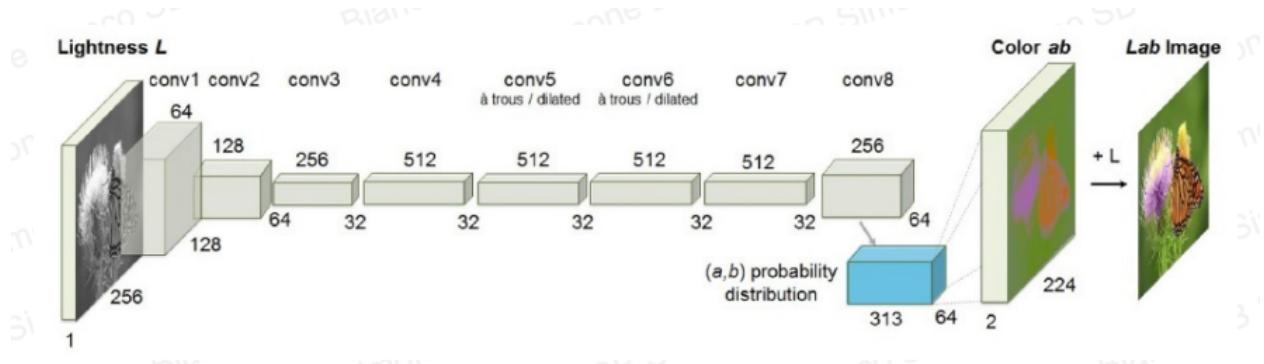
Image generation with Inpainting



Questa non è una task semplice, il modello deve imparare molte informazioni.

Image generation with Colorization

L'idea è che ci sono degli spazi colori diversi, tipo YCbCr...



Qui iniziamo con un depth channel, e una risoluzione alta. Riduciamo la riduzione spaziale e aumentiamo quella di profondità. Poi ad un certo punto manteniamo la stessa dimensione e applichiamo solo la non linearità. Facciamo questo perchè vogliamo ricostruire i dettagli della nostra immagine, se riduciamo troppo la dimensione spaziale, questo è utile per una task di classification, ma se vogliamo tornare all'immagine originale non dobbiamo ridurre troppo la dimensione spaziale.

Poi iniziamo ad aumentare la dimensione spaziale, nello stesso modo in cui usavamo la transposed convolution nelle GANs.

Context-based image feature learning

Le pretext tasks basate sul contesto utilizzano principalmente le features di contesto delle immagini, inclusa la similarità contestuale, la struttura spaziale e la struttura temporale come segnali di supervisione.

Le caratteristiche vengono apprese dalle CNN attraverso il processo di risoluzione delle pretext tasks, progettate sulla base degli attributi del contesto delle immagini.

Abbiamo 2 modi di usare questa context-similarity. Possiamo formulare il problema come una **predictive task** o come una **contrastive task**.

In entrambi i metodi, i dati vengono prima raggruppati in diversi gruppi, assumendo che i dati appartenenti allo stesso gruppo abbiano un'alta similarità contestuale, mentre i dati di gruppi diversi abbiano una bassa similarità contestuale.

I **task predittivi** implicano l'addestramento di reti per predire l'ID del gruppo dei dati, solitamente con una cross entropy loss.

Le task di contrasto implicano l'addestramento di reti per minimizzare direttamente le distanze delle caratteristiche all'interno dello stesso gruppo e massimizzare le distanze delle caratteristiche tra gruppi diversi, solitamente con una triplet loss o una contrastive loss.

Learning with context similarity: predictive

Questa è una possibilità per la predictive task.

Nell'SSL, i metodi di clustering sono utilizzati principalmente come uno strumento per raggruppare i dati delle immagini.

Un metodo semplice consisterebbe nel raggruppare i dati delle immagini basandosi su caratteristiche progettate manualmente, come SIFT seguite da BoW.

Dopo il clustering, si ottengono diversi cluster, mentre le immagini all'interno di un cluster hanno una distanza minore nello spazio delle caratteristiche, mentre le immagini di cluster diversi presentano una distanza maggiore nello spazio delle caratteristiche. Più piccola è la distanza nello spazio delle caratteristiche, più simile è l'immagine nell'aspetto nello spazio RGB.

Successivamente, una CNN può essere addestrata per classificare i dati utilizzando l'assegnazione del cluster come pseudo-label di classe.

Per fare questa task, la CNN deve imparare l'invarianza all'interno di una classe e la varianza tra classi diverse, quindi deve imparare il significato semantico delle immagini.

Learning with context similarity: contrastive

Un altro modo di sfruttare la similarità contestuale per l'SSL è il **contrasting**.

L'idea generale è quella di trainare reti per massimizzare l'agreement di diverse viste della stessa scena, minimizzando allo stesso tempo l'accordo tra viste di scene differenti.

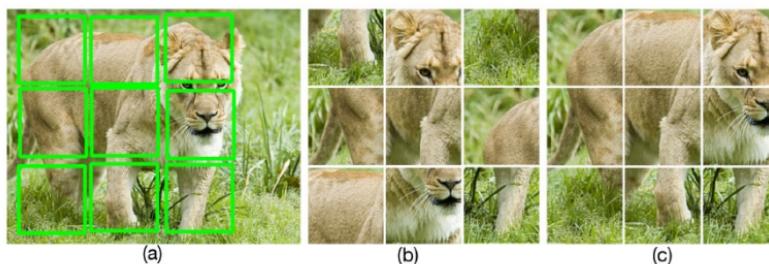
Il metodo recente allo stato dell'arte è SimCLR, che apprende caratteristiche confrontando immagini dopo una composizione di tecniche di data augmentation.

Le coppie positive sono costruite campionando due immagini ottenute applicando diverse tecniche di data augmentation alla stessa immagine, mentre le coppie negative includono due immagini differenti.

Questo metodo supera significativamente altri metodi di apprendimento auto-supervisionato sul dataset ImageNet.

Learning with spatial context structure

- Images contain rich spatial context information such as the relative positions among different patches from an image, which can be used to design the pretext task for SSL.
- The pretext task can be to predict the relative positions of two patches from same image, or to recognize the order of a shuffled sequence of patches from same image.
- The context of full images can also be used as a supervision signal to design pretext tasks such as to recognize the rotating angles of the whole images.
- To accomplish these pretext tasks, CNNs need to learn spatial context information such as the shape of the objects and the relative positions of different parts of an object.
- Following this idea, more methods are proposed to learn image features by solving more difficult spatial puzzles.
- One typical work attempted to solve an image Jigsaw puzzle with CNNs.



- The shuffled image patches are fed to the network which is trained to recognize the correct spatial locations of the input patches by learning spatial context structures of images such as object color, structure, and high-level semantic information.

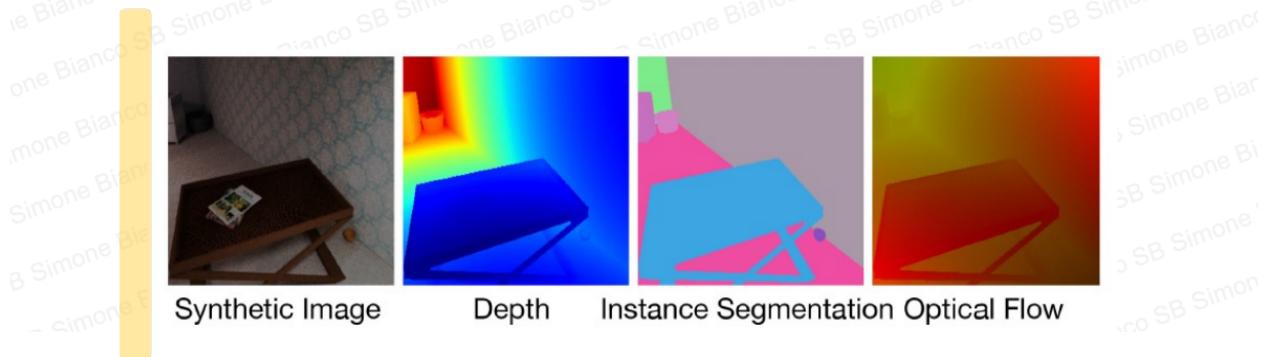
- Given 9 image patches from an image, there are 362,880 (=9!) possible permutations and a network is very unlikely to recognize all of them because of the ambiguity of the task.
- To limit the number of permutations, usually, Hamming distance is employed to choose only a subset of permutations among all the permutations, i.e., those with a relative large Hamming distance.
- Only the selected permutations are used to train CNN to recognize the permutation of shuffled image patches.
- The main principle of designing puzzle tasks is to find a suitable task which is not too difficult and not too easy for a network to solve:
 - If it is too difficult, the network may not converge due to the ambiguity of the task.
 - If it is too easy, it can easily learn trivial solutions.

Free semantic label-based image feature learning

- The free semantic label refers to labels with semantic meanings that are obtained without involving any human annotations.
- Generally, the free semantic labels such as segmentation masks, depth images, optical flows, and surface normal images can be rendered by game engine or generated by hard-code methods.
- Since these semantic labels are automatically generated, the methods using the synthetic datasets or using them in conjunction with a large unlabeled image or video datasets are considered as self-supervised learning methods.

Per esempio per generare delle immagini in un game engine, il cui risultato è la prima immagine, bisogna avere molte informazioni come l'instance segmentation perché ogni oggetto è in una posizione e ha delle proprietà per esempio dei materiali...

- Given models of various objects and layouts of environments, game engines are able to render realistic images and provide accurate pixel-level labels.
- Since game engines can generate large-scale datasets with negligible cost, various game engines (e.g., Airsim, Carla, etc.) have been used to generate large-scale synthetic datasets with high-level semantic labels including depth, contours, surface normal, segmentation mask, and optical flow for training deep networks.



Quindi partendo dalla prima immagine posso trainare una rete per imparare quelle a destra, dato che le posso prendere dal game engine.

- Game engines can generate realistic images with accurate pixel-level labels with very low cost.
- However, due to the domain gap between synthetic and real-world images, the CNNs purely trained on synthetic images cannot be directly applied to real-world images.
- To utilize synthetic datasets for self-supervised feature learning, the domain gap needs to be explicitly bridged.
- In this way, the CNN trained with the semantic labels of the synthetic dataset can be effectively applied to real-world images.

Learning with labels generated by hard-code programs

- Applying hard-code programs is another way to automatically generate semantic labels such as salience, foreground masks, contours, depth for images and videos.
- With these methods, very large-scale datasets with generated semantic labels can be used for self-supervised feature learning.
- This type of methods generally has two steps:
 - 1) label generation by employing hard-code programs on images or videos to obtain labels,
 - 2) train CNNs with the generated labels.
- Various hard-code programs have been applied to generate labels for self-supervised learning methods including methods for foreground object segmentation, edge detection, and relative depth prediction.

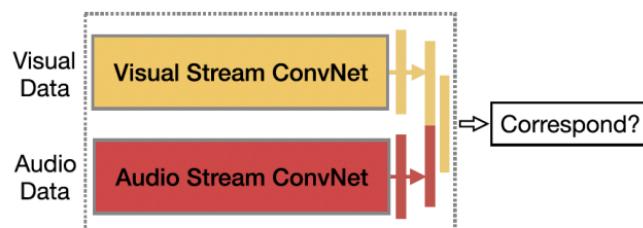
Cross modal-based learning

- Cross modal-based learning methods usually learn features from the correspondence of multiple data streams including RGB frame sequence, optical flow sequence, audio data, and camera pose.
- In addition to rich temporal and spatial information in videos, optical flow sequence can be generated to specifically indicate the motion in videos, and the difference of frames can be computed with negligible time and space-time complexity to indicate the boundary of the moving objects.
- Similarly, audio data also provide a useful hint about the content of videos.

- Based on the type of data used, these methods fall into three groups:
 - methods that learn features by using the RGB and optical flow correspondence
 - methods that learn features by utilizing the video and audio correspondence
 - ego-motion that learn by utilizing the correspondence between egocentric video and egomotor sensor signals.

- Usually, the network is trained to recognize if the two kinds of input data are corresponding to each other, or is trained to learn the transformation between different modalities

Visual Audio Correspondence Network



Performance comparison

L'unico modo di farlo, è testare su una serie in comune di downstream tasks.

L'idea è che il tipo di features che saranno molto buone per una serie di downstream tasks saranno le migliori per un altro tipo di task.

Una di queste downstream tasks è la classificazione su imagenet.

All'inizio abbiamo detto che le performance di questo tipo di metodi raggiungono e in alcuni casi riescono anche a superare le performance che vengono raggiunte da alcuni metodi supervised.

L'idea è che quei abbiamo networks molto grandi che possono essere trainati su una quantità molto grande, e sono modelli talmente grandi che non possono overfittare, e sono molto robusti.

Queste features sono così robuste e così generali (perchè hanno visto tante immagini). Questo è quello che c'è alla base dei foundational models, molti sono trainati in un metodo self-supervised.