

MODULI FORMATIVI ESTATE 2022

INTRODUZIONE ALL'INTELLIGENZA ARTIFICIALE

Computer Vision e Reti Neurali

Lezione III – 31 agosto 2022



ARTIFICIAL INTELLIGENCE
& DATA ANALYTICS

PRESENTAZIONI



MARCO ZULLICH

Dottorando in Ingegneria Industriale e dell'Informazione
Docente di informatica e Deep Learning

...e voi?

OUTLINE DEL CORSO (I)

PARTE I – Computer Vision classica

90 min circa

Introduzione alla visione artificiale e alle immagini digitali

- Momento interattivo: gli spazi di colore

Le feature e la *Vision* classica

- Convoluzione e filtri
- *Feature Detection* classica
- Concetto di punti chiave

OUTLINE DEL CORSO (II)

PARTE II – Reti Neurali orientate alla visione artificiale

Dal modello lineare alla rete neurale

90 min circa

- Momento interattivo: rete neurale per il riconoscimento di cifre

Approfondimenti sulle tecniche delle reti neurali

45 min circa

- Funzioni di attivazione e ottimizzazione

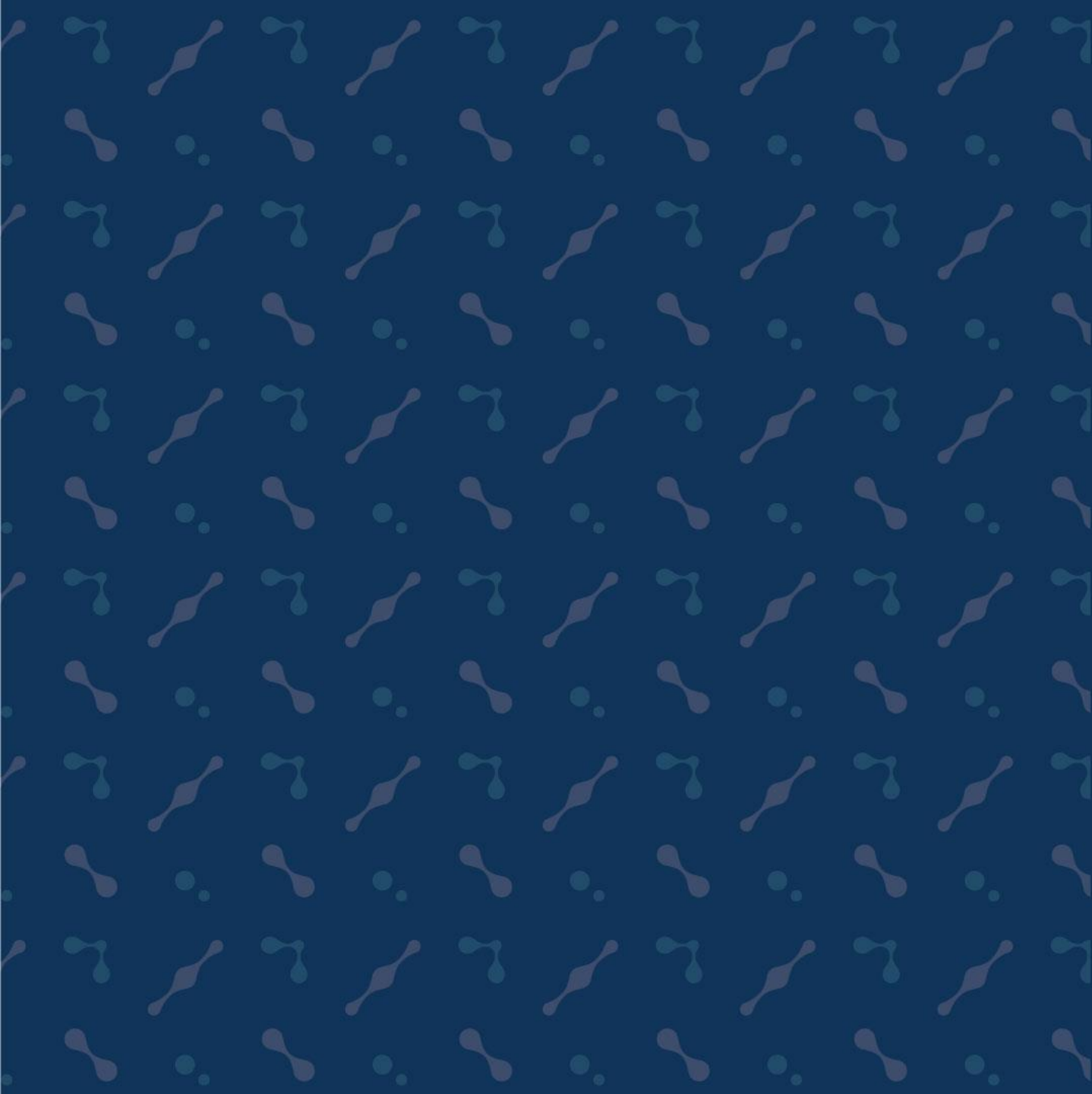
Le reti neurali convoluzionali e applicazioni alla visione

60 min circa

- Esempi applicativi

MATERIALE DELLA LEZIONE

- Notebook e materiale interattivo qui: bit.ly/AI_22_intro
- Verranno caricate anche queste slide dopo la lezione



.01

Introduzione alla vision artificiale e alle immagini digitali

COSA VEDETE IN QUESTA IMMAGINE?



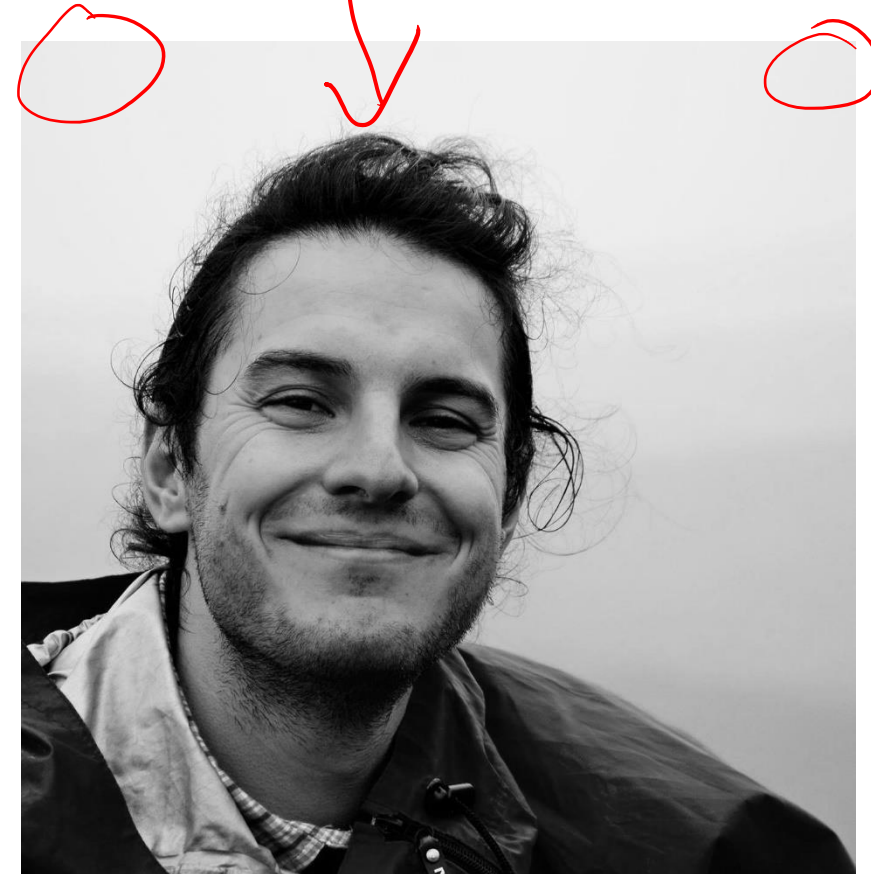
... E IN QUESTA IMMAGINE?



CHE COSA VEDE UN COMPUTER

PIXEL

234	235	236	236	237	237	237	238	238	239	238	239	239	239	239	239	239	239	239	239	238	238	238	238	237	237	237	237	236	235	235	234
234	235	236	237	237	237	237	238	238	239	239	239	239	239	239	239	239	239	239	239	239	238	238	237	237	237	237	236	235	234	234	
234	235	236	236	237	237	237	238	238	239	239	239	239	239	239	239	239	239	239	239	239	238	238	237	237	237	236	236	235	234	234	
234	235	236	236	237	236	237	238	238	239	239	197	85	217	240	214	226	239	239	239	239	238	237	237	237	237	236	236	235	234	233	
233	234	235	236	237	237	237	236	230	152	62	38	89	38	46	114	133	209	222	238	238	237	237	237	237	236	235	235	234	233		
233	234	235	236	236	237	236	219	127	16	17	13	7	6	18	35	87	216	238	238	237	237	237	237	236	235	235	234	233	233		
233	234	234	235	236	237	235	210	16	2	13	20	16	19	2	22	38	48	204	238	237	238	237	236	236	235	234	233	233	232		
233	233	234	234	235	236	180	16	48	185	192	183	182	163	61	7	2	32	171	208	228	234	233	233	233	231	231	228	228	227		
233	233	233	235	235	230	16	46	174	192	192	192	191	185	179	170	61	29	187	227	222	231	231	231	230	228	229	227	226	226		
229	229	229	229	230	221	14	63	173	176	177	182	180	161	172	171	129	31	217	223	223	230	230	230	230	229	228	227	225	225		
227	227	228	228	228	33	31	107	167	185	182	181	185	179	189	173	150	21	225	223	228	227	227	225	225	223	223	220	218	219		
220	222	226	225	210	12	10	172	58	26	19	73	147	164	143	155	161	16	207	221	222	222	221	219	221	220	219	219	217	216		
212	213	216	216	149	2	17	170	100	40	62	53	163	126	4	6	81	12	219	215	222	221	220	220	219	218	216	215	216	213		
208	211	211	209	82	4	91	190	162	133	92	116	168	139	13	58	80	34	2	210	214	217	219	219	217	216	217	215	213	212		
207	207	206	207	85	90	152	174	184	178	169	161	174	139	79	63	146	8	167	195	204	212	215	214	212	211	209	206	202	203		
204	204	205	208	157	13	105	170	173	171	124	165	165	157	163	187	185	9	199	181	201	204	205	204	205	203	203	201	199	199		
201	202	203	201	153	162	141	150	100	51	123	75	75	6	114	112	146	30	201	198	195	200	201	202	202	202	203	201	200	199	199	
198	199	199	176	17	10	144	124	93	99	134	107	99	105	118	32	67	149	128	194	202	202	202	202	202	200	200	199	196	195		
195	198	198	197	108	0	120	105	130	142	144	150	142	115	75	105	84	198	192	194	201	201	200	200	200	199	198	197	196	194		
9	10	8	4	203	7	29	77	87	154	143	106	76	107	126	101	78	189	197	199	199	199	199	199	199	198	197	195	194	194	191	
9	7	3	224	210	0	148	51	89	113	167	154	128	151	132	33	159	192	195	195	196	196	196	196	196	195	194	194	191	191	190	
12	121	211	202	192	2	153	28	80	54	112	82	112	107	63	35	178	190	192	192	194	193	193	194	193	191	193	191	190	190		
25	185	128	191	197	88	138	104	25	5	28	13	17	15	5	47	27	117	110	185	187	185	186	186	188	190	190	189	187	186		
55	45	138	188	180	207	110	114	63	28	29	12	8	18	8	12	69	34	86	98	101	163	178	178	177	178	175	177	179	180		
59	39	90	159	152	147	48	105	75	63	58	10	26	73	11	14	75	18	60	78	83	96	121	171	172	173	172	170	169	168		
16	23	24	125	174	147	129	64	91	74	49	49	63	10	14	18	21	13	20	44	62	82	99	84	148	173	171	168	167	165		
25	18	19	164	168	162	185	131	65	81	57	30	7	19	24	17	20	65	12	22	27	44	65	95	77	96	169	167	167	163		
28	21	16	16	141	136	176	166	182	63	65	39	23	23	4	0	16	27	27	18	26	27	51	61	65	10	23	167	164			
17	22	16	15	189	191	107	103	27	72	67	158	17	3	2	35	27	52	35	28	21	33	34	26	40	16	13	9	9	165		
11	20	24	18	29	187	190	156	118	60	8	7	28	25	26	40	10	4	4	11	22	22	27	26	17	7	10	10	14	21		

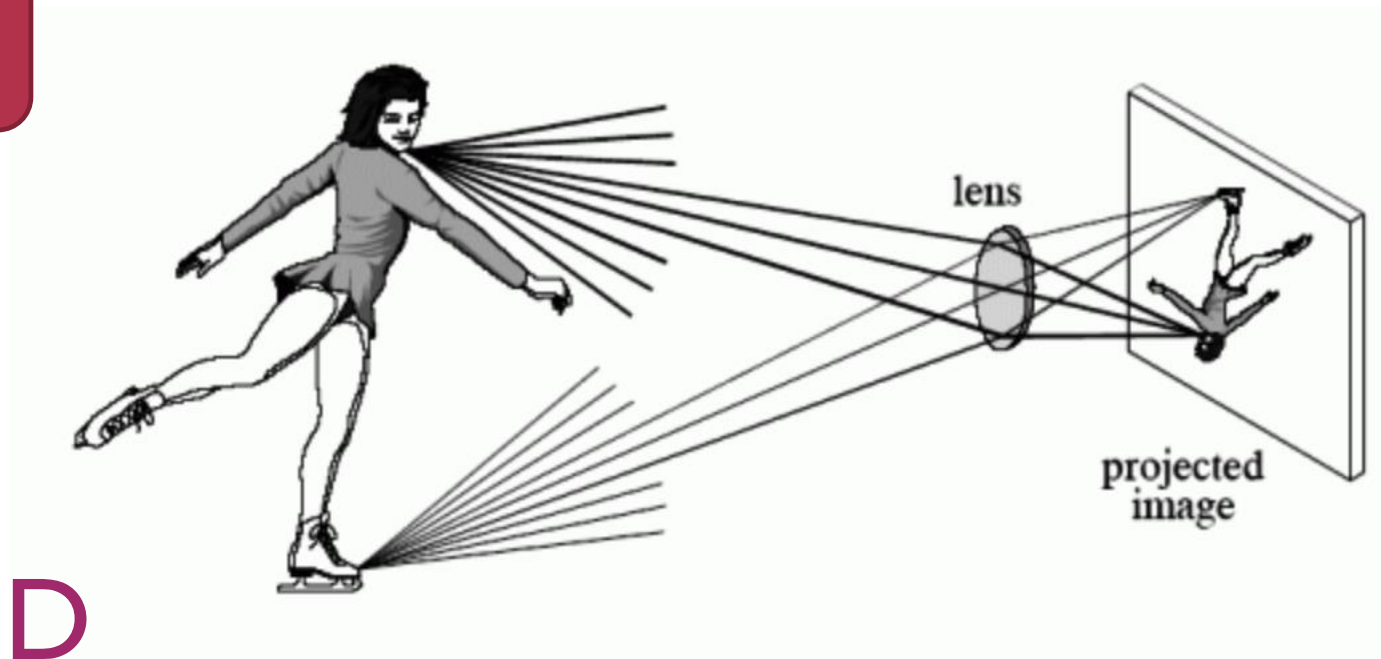


COMPUTER VISION O VISIONE ARTIFICIALE

La **visione artificiale** (*Computer Vision*, CV) è la disciplina che si occupa di permettere ad una macchina di «vedere».

DIFFICOLTÀ DELLA VISIONE ARTIFICIALE – DOVE

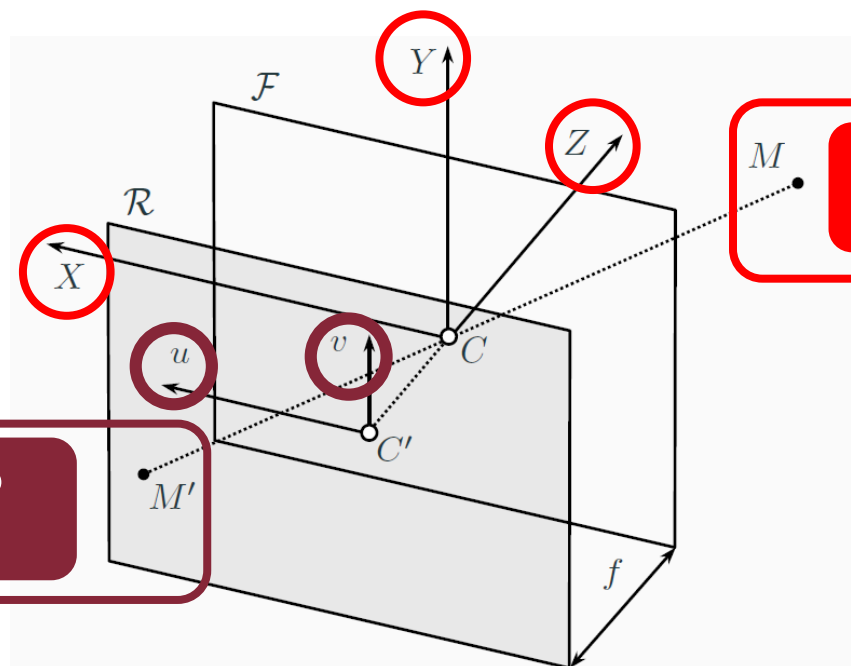
«WHERE»



3D → 2D

PERDITA DI PROFONDITÀ

Scattare una fotografia equivale a PROIETTARE il mondo tridimensionale in uno spazio a bidimensionale, perdendo di fatto l'informazione sulla profondità

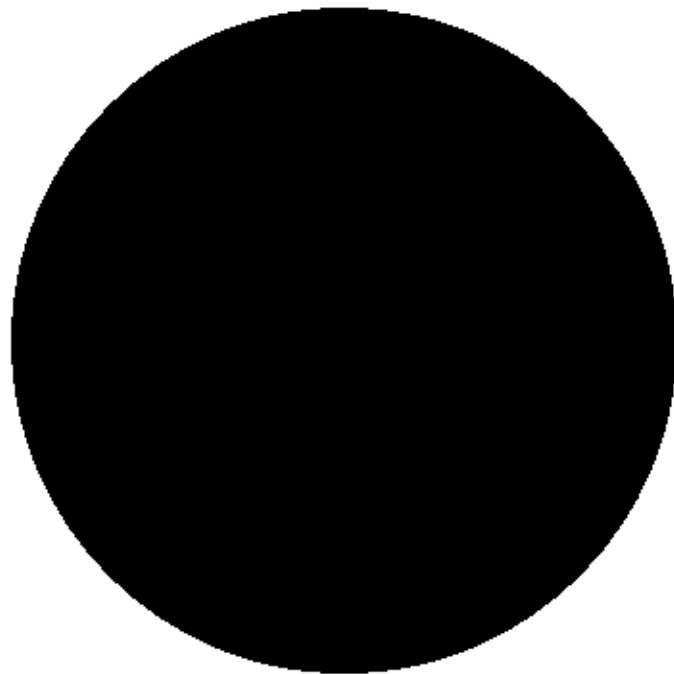
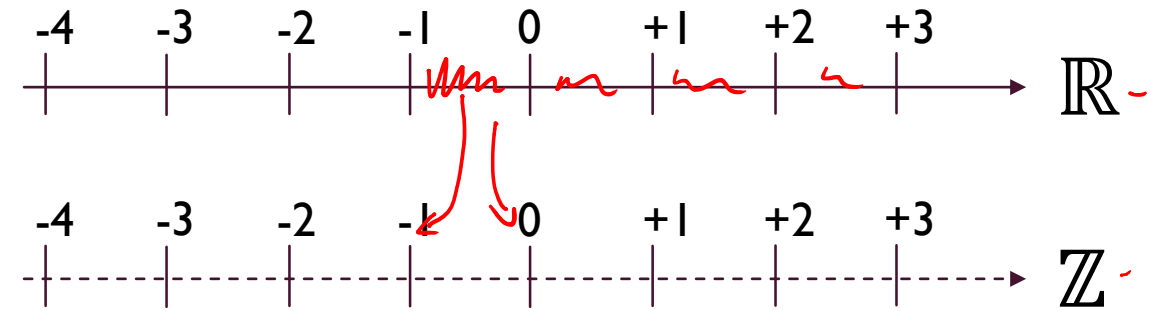


Oggetto nel mondo reale (3D)

Proiezione sul piano d'immagine (2D)

QUANTIZZAZIONE DELLO SPAZIO

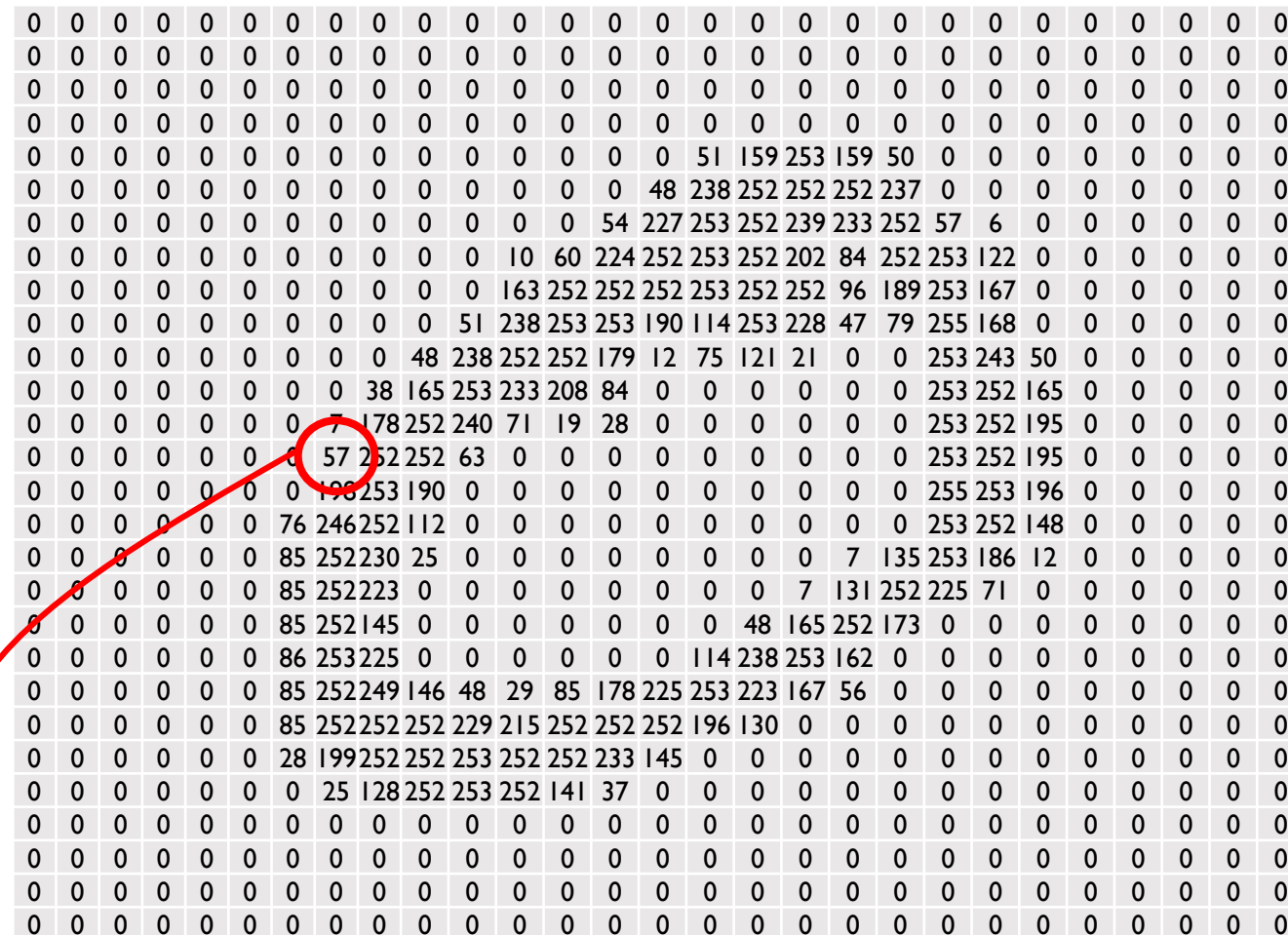
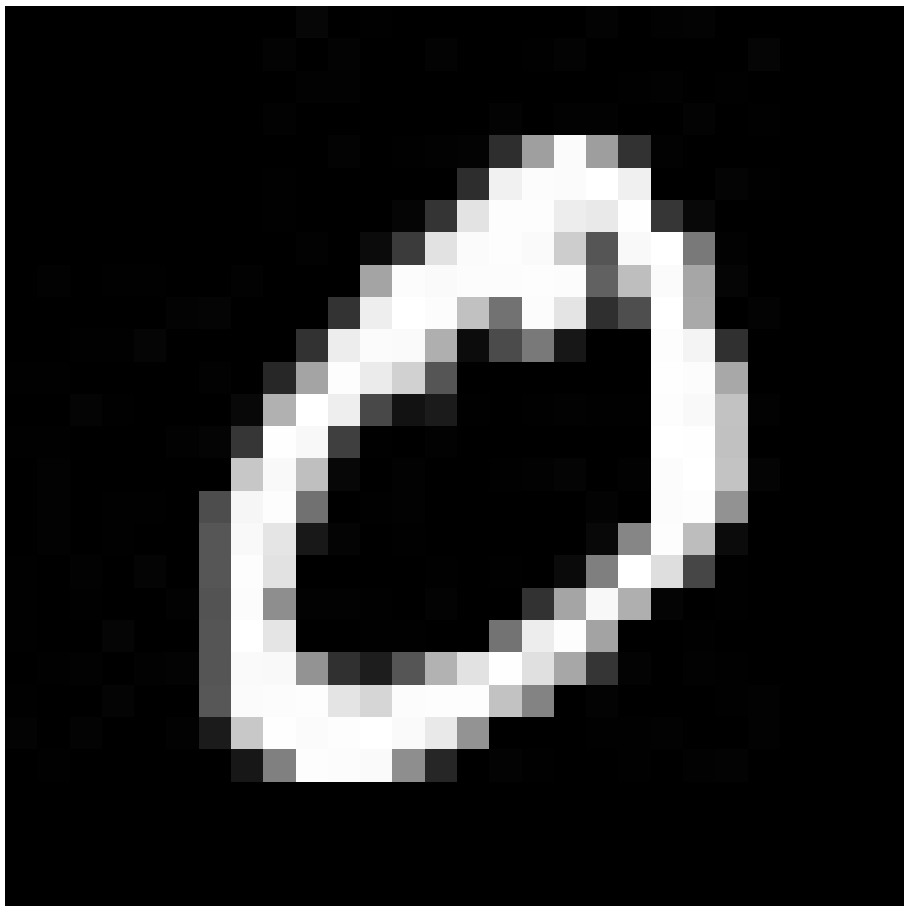
- Dimensioni spaziali → CONTINUE
- Immagine digitale → DISCRETA
 - Il quanto è il PIXEL



VS.



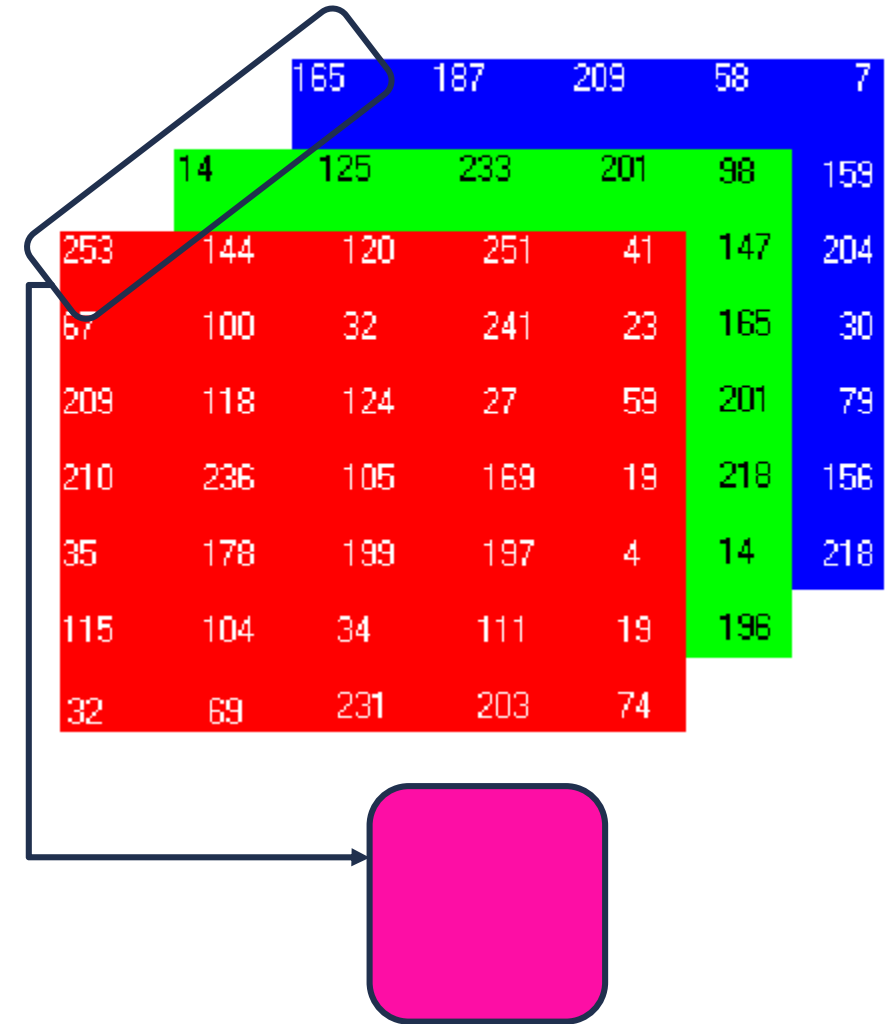
CODIFICA DELL'IMMAGINE (SCALA DI GRIGI)



Valore $\equiv$ Intensità

CODIFICA DELL'IMMAGINE (COLORI)

- Numerosi paradigmi
- Il più conosciuto è il RGB (Red / Green / Blue)
- L'immagine è codificata in 3 canali
- Un pixel viene codificato secondo 3 numeri diversi
- Ogni numero rappresenta l'intensità del singolo canale (da 0 a 255)
- La sovrapposizione dei 3 canali dà origine al colore così come percepito dall'occhio umano



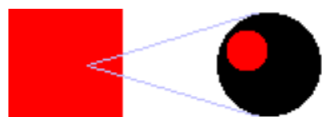
COMEVIENE RESO IL COLORE IN RGB?

Tratto da https://www.chem.purdue.edu/gchelp/cchem/RGBColors/body_rgbcolors.html

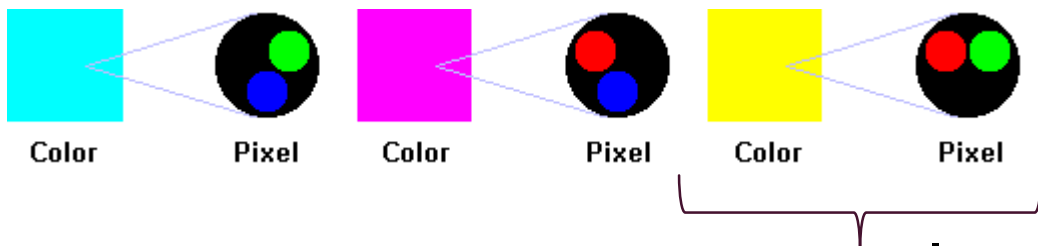
I monitor sono specializzati a rappresentare i colori secondo la codifica RGB



Ogni pixel è in realtà composto da tre piccoli puntini che riproducono il colore rosso, verde, blu



È banale mostrare il colore rosso (o verde o blu)



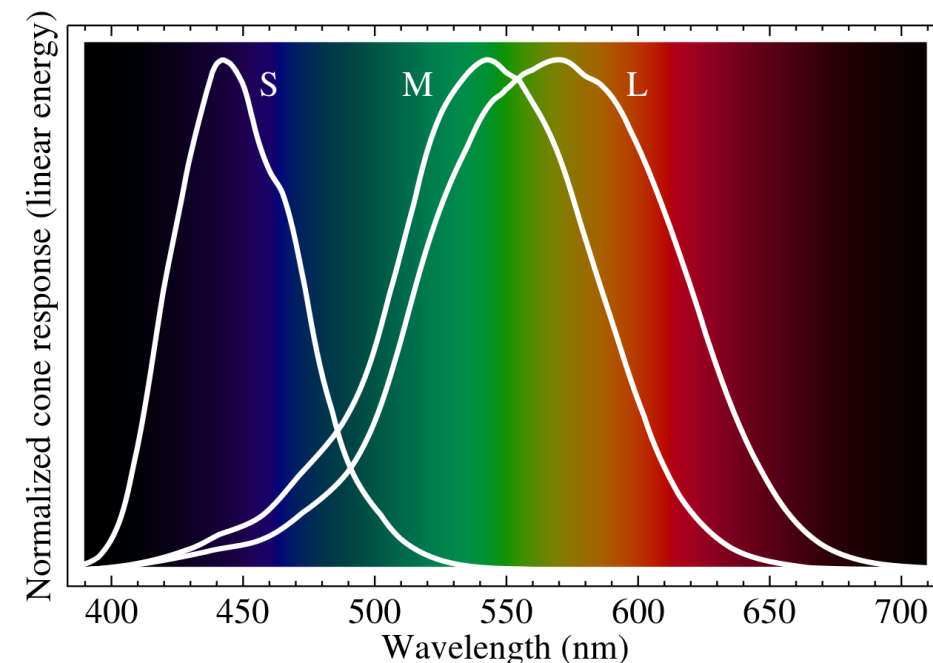
Gli altri colori vengono riprodotti «accendendo» i relativi puntini RGB dell'intensità dettata dalla codifica RGB

Es. **GIALLO** = (255, 255, 0)
R G B

I puntini corrispondenti a Rosso e Verde si accendono al massimo dell'intensità, il Blu rimane spento

PERCHÉ RGB?

- Il paradigma RGB è utilizzato in quanto cerca di riprodurre il meccanismo della visione dei colori umana
- Nella retina umana si trova un insieme di cellule chiamate **coni**
- I coni si dividono in tre tipologie, ognuna delle quali è specializzata a percepire uno specifico *range* di colori
- Il picco di queste curve si trova all'incirca in corrispondenza dei colori blu, verde, rosso
- Il cervello si occupa poi di combinare queste tre diverse percezioni in un colore singolo

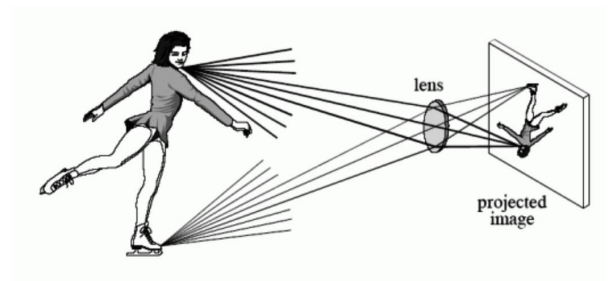
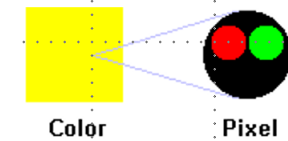


Di BenRG - Opera propria, Pubblico dominio,
<https://commons.wikimedia.org/w/index.php?curid=7873848>

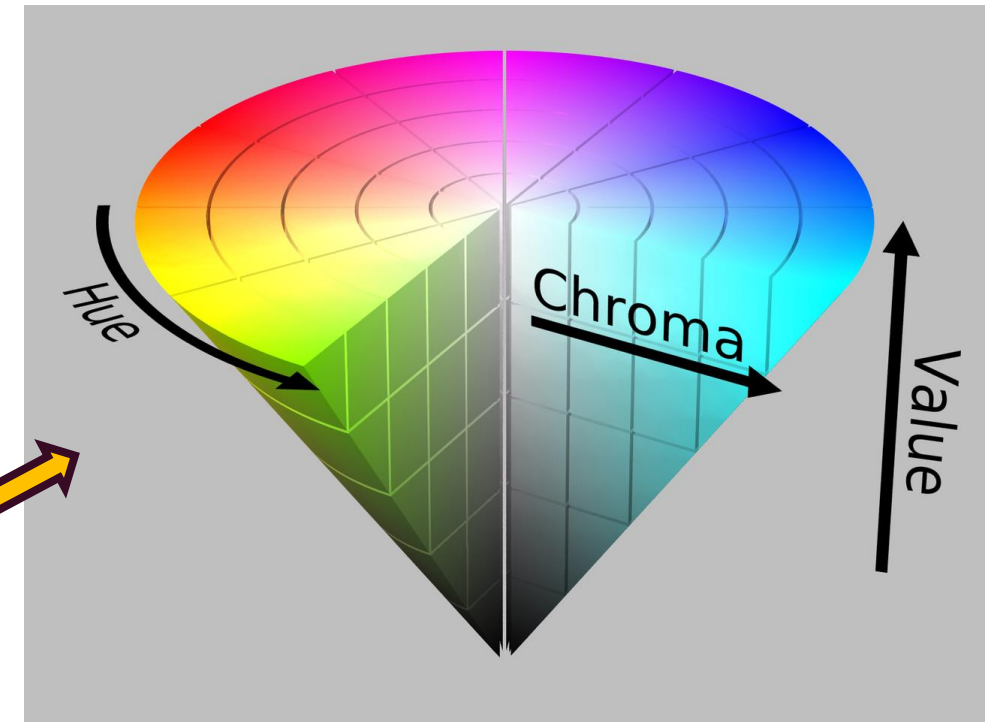
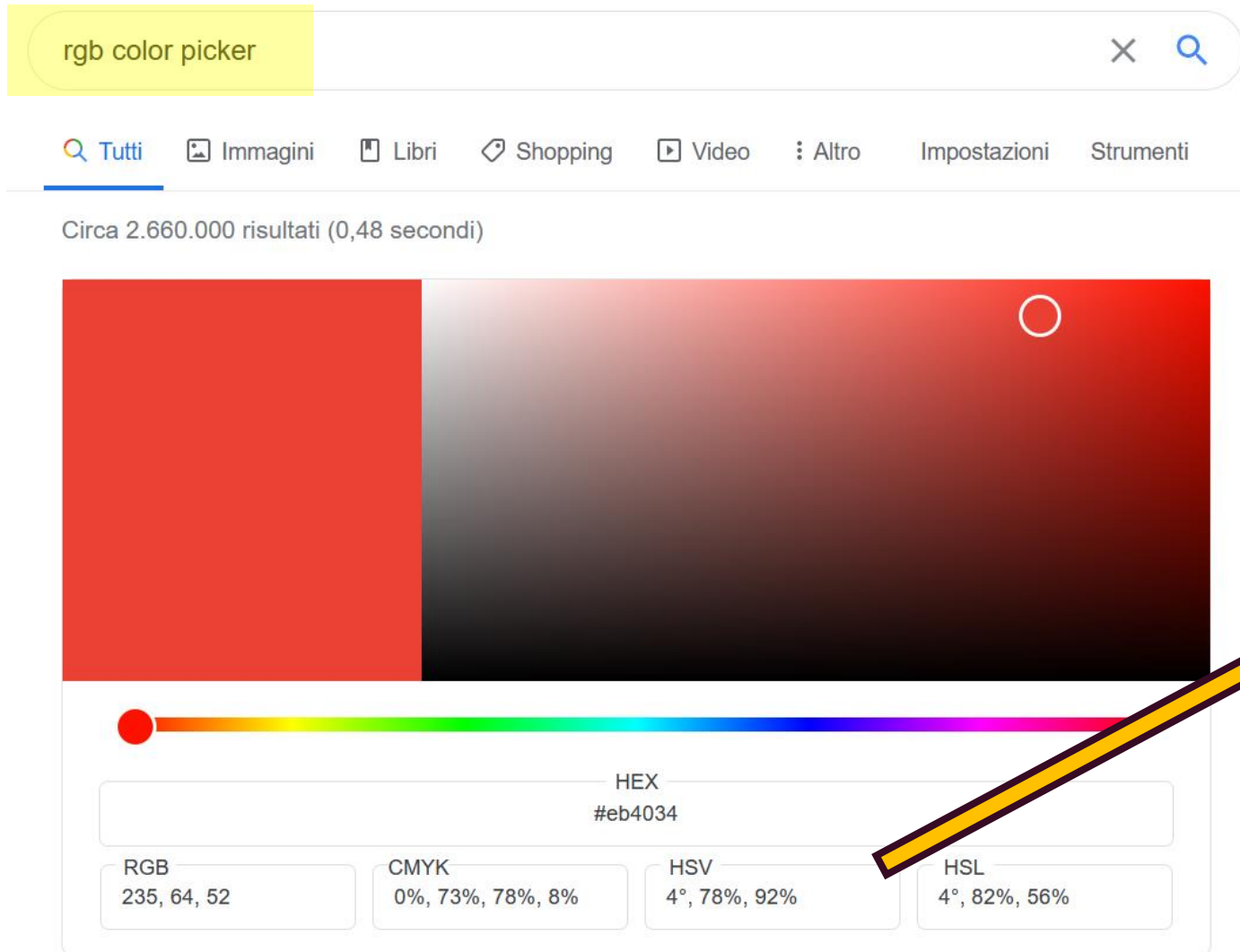
RIASSUMENDO

- Un computer «vede» un'immagine a colori come **3 «griglie» bidimensionali di pixel**
- Ogni griglia contiene **valori di intensità** di 3 colori fondamentali: rosso, verde e blu
- (Codifica RGB): simile alla *percezione cromatica* umana
- Tramite questa codifica, si possono visualizzare quasi tutti i colori dello spettro visibile
- I monitor sono composti di pixel in grado di combinare le diverse intensità dei tre colori fondamentali a creare tutti gli altri colori
- Scattare un'immagine = proiezione 3D $\rightarrow$ 2D, si perde la percezione della profondità
- Esistono tecniche per «recuperare» la profondità andando a «combinare» scatti multipli dello stesso oggetto da diverse prospettive

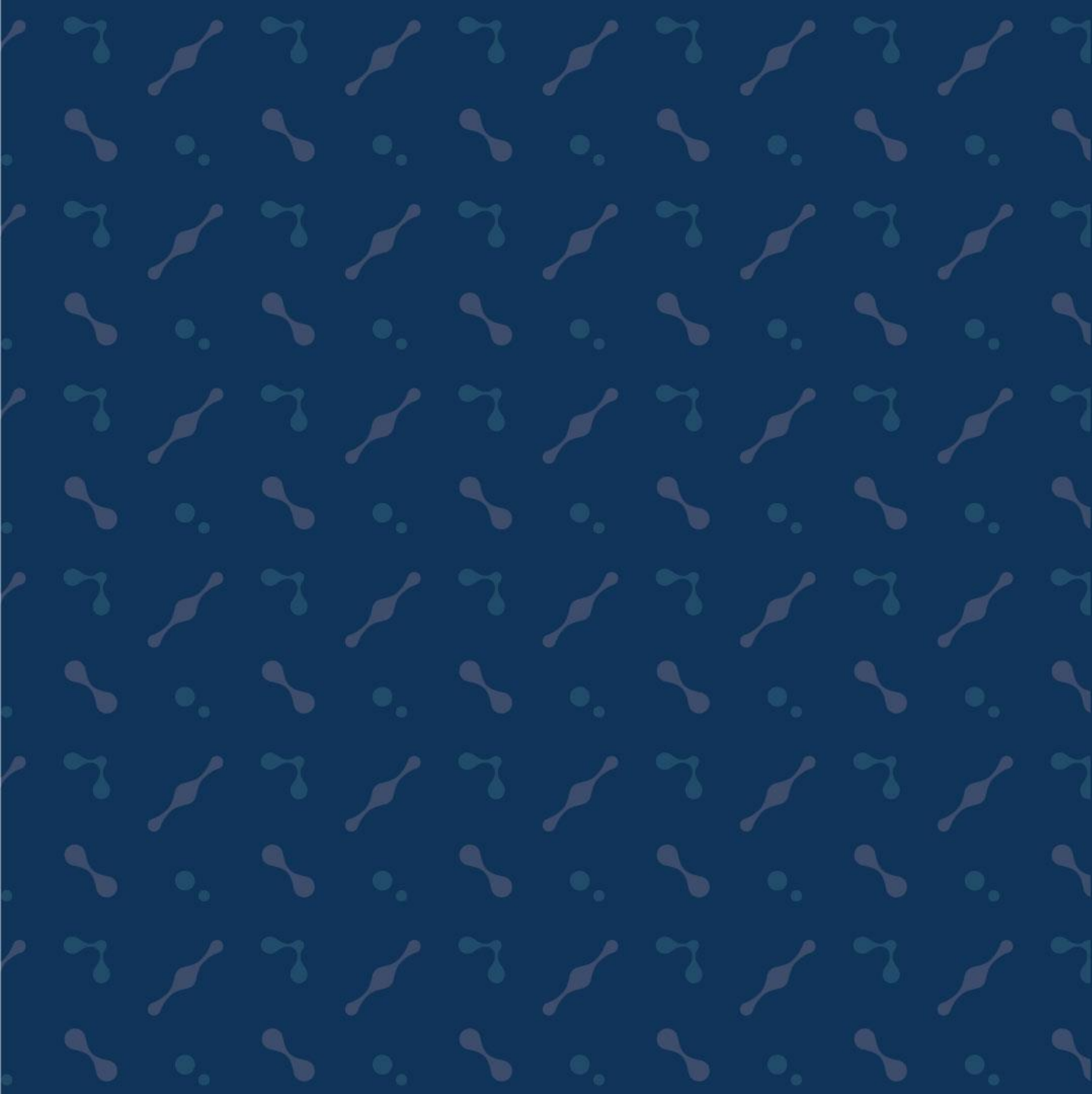
		165	187	209	58	7
	14	125	233	201	98	159
253	144	120	251	41	147	204
67	100	32	241	23	165	30
209	118	124	27	59	201	79
210	236	105	169	19	219	156
35	178	199	197	4	14	218
115	104	34	111	19	196	
32	69	231	203	74		



MOMENTO INTERATTIVO (I) - COLORSPACES



Credits: Jacob Rus, SharkD, distributed under CC-BY 3.0 license



.02

**Le *feature* e la *vision*
classica**

DIFFICOLTÀ DELLA VISIONE ARTIFICIALE – CHE COSA



Illumination



Occlusion



Deformation

«WHAT»



Background

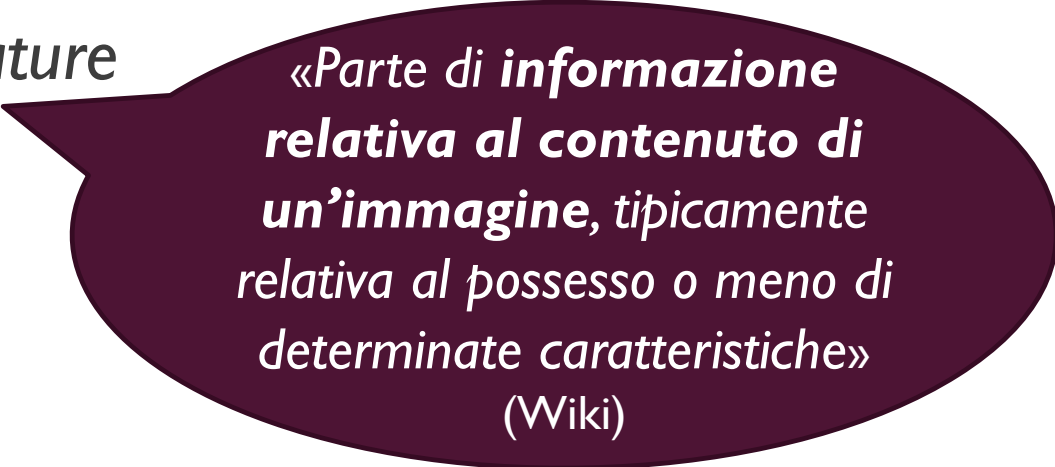


Intraclass variation

IL CONCETTO DI FEATURE

- Il «che cosa» si basa sul concetto di *feature*

- Caratteristica
- Componente
- ...



«Parte di **informazione** relativa al contenuto di **un'immagine**, tipicamente relativa al possesso o meno di determinate caratteristiche»
(Wiki)

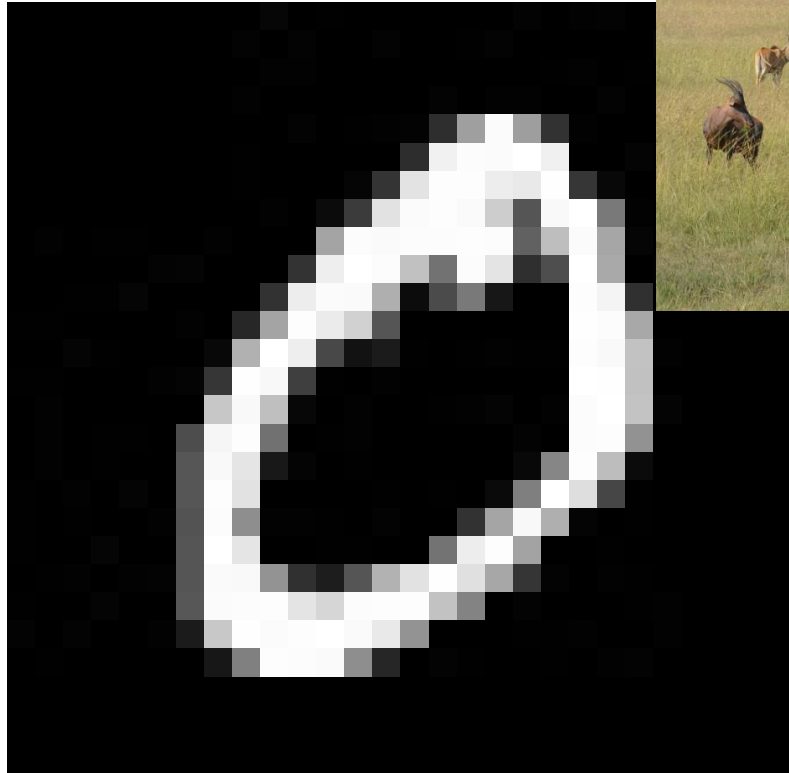
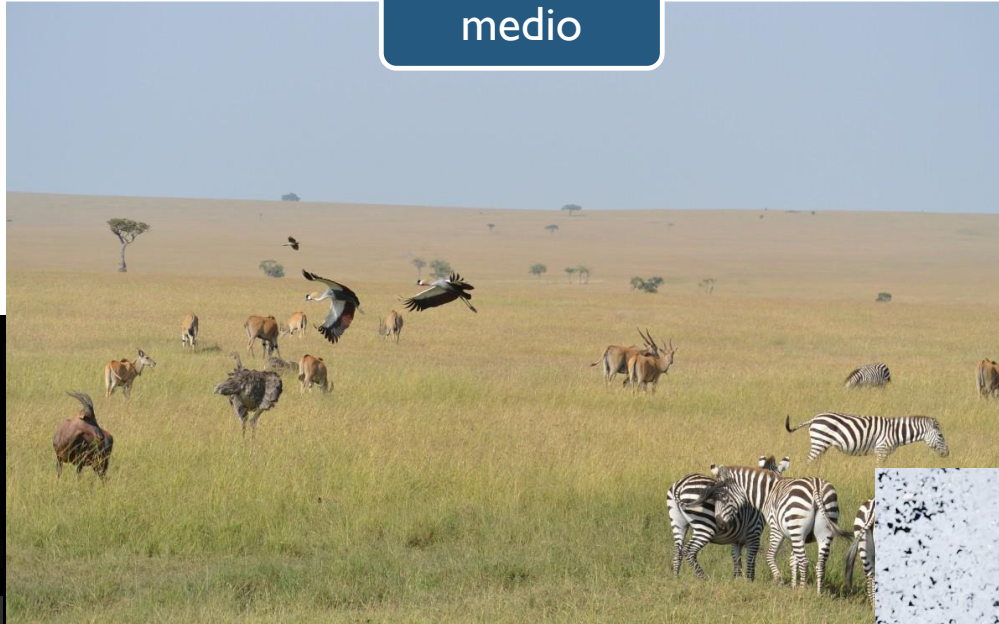
- Le feature sono **ordinabili** in maniera gerarchica
- L'ordine dipende dalla *vicinanza* della feature alla rappresentazione *matriciale* dell'immagine

RIPRENDENDO LE IMMAGINI PRECEDENTI...

basso

medio

alto

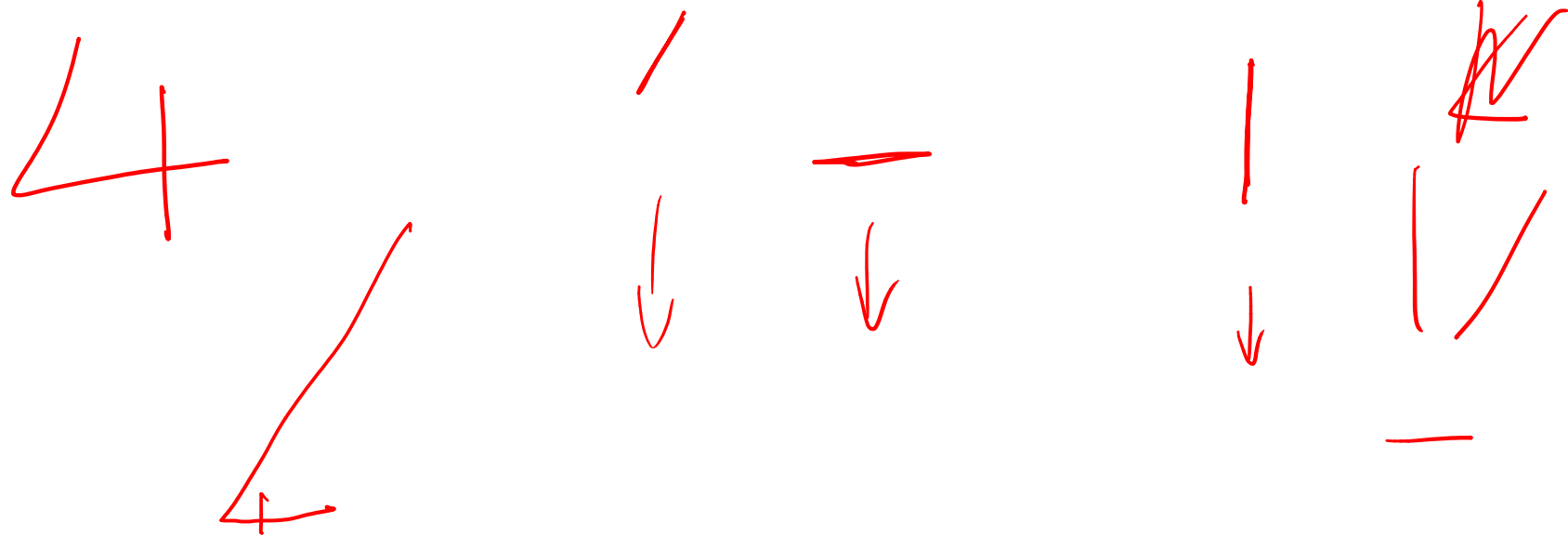


FEATURE DI BASSO LIVELLO

- Colore
- Bordi
 - Linee
 - Curve
 - Orientamento

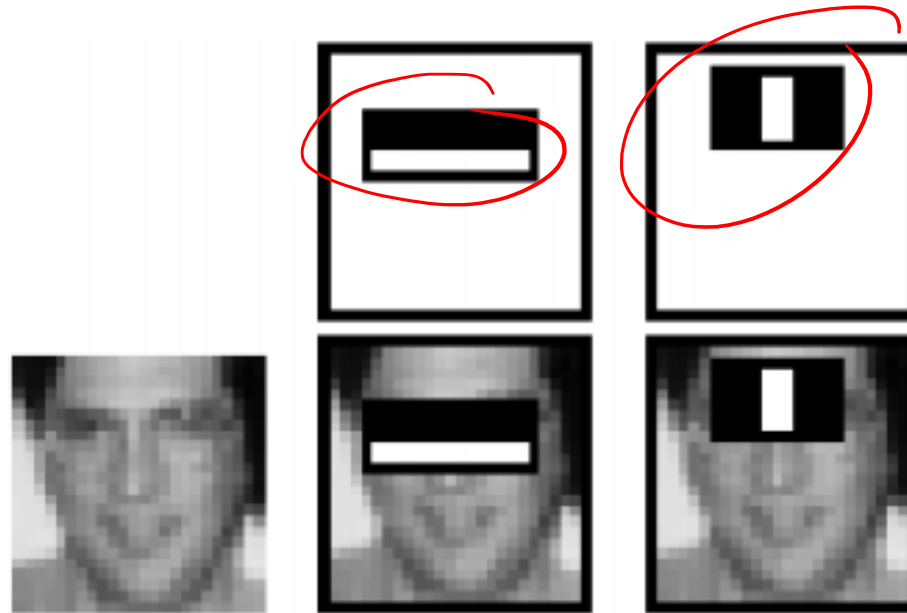
COMBINAZIONE DI FEATURE

- La combinazione di feature di basso livello permette di ottenere feature di livello più alto.
- Esempio:



UN ESEMPIO PIÙ «ACCADEMICO»

Riconoscimento di volti (Viola & Jones)



FEATURE DETECTION

- Feature detection → RICONOSCIMENTO DI CARATTERISTICHE
- Compito estremamente difficile per un computer
- Storicamente conseguito (con risultati «altalenanti») tramite l'utilizzo di FILTRI & CORRELAZIONE / CONVOLUZIONE

LAVAGNA INTERATTIVA: CROSS-CORRELAZIONE

RICONOSCIMENTO DI 1 CARATTERISTICO

I

0	1	5	0	3
3	2	3	1	2
4	3	1	2	1

6x5

F

0	1	2
2	1	2
1	2	3

I

1	5	0
2	7	1
3	1	2

F

0	1	2
2	1	2
1	2	3

0	5	0
4	7	2
3	2	6

Σ

$\rightarrow$

~~35~~ 29

I * F

3

46	29	x

4

Mappa che identifica la "risposta" del filtro rispetto all'immagine

FILTRO MEDIO («BOX»)

- Il filtro medio utilizza un *filtro* kernel quadrato di lato n (dispari)
- Ogni elemento del kernel è $1/n^2$

F

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

1	1	1
1	1	1
1	1	1

I

0	1	2
1	2	2
3	2	1

$$0 \cdot \frac{1}{9} + 1 \cdot \frac{1}{9} + 2 \cdot \frac{1}{9} + \dots$$

$$\frac{0 \cdot 1 + 1 \cdot 1 + 2 \cdot 1 + \dots + 1 \cdot 1}{9}$$

$$= \frac{0 + 1 + 2 + 1 + 2 + 2 + 2 + 1}{9}$$

$I \star F$ → valore medio del pixel
in $I[1:3; 1:3]$

→ $I[2:4, 1:3]$

IL FILTRO MEDIO (2)

- Il filtro medio ha come risultato quello di sostituire il pixel centrale con la media del suo vicinato
- Maggiore è la grandezza del filtro, maggiore è l'«**effetto media**», che risulta in una **sfocatura**



Filtro 5x5



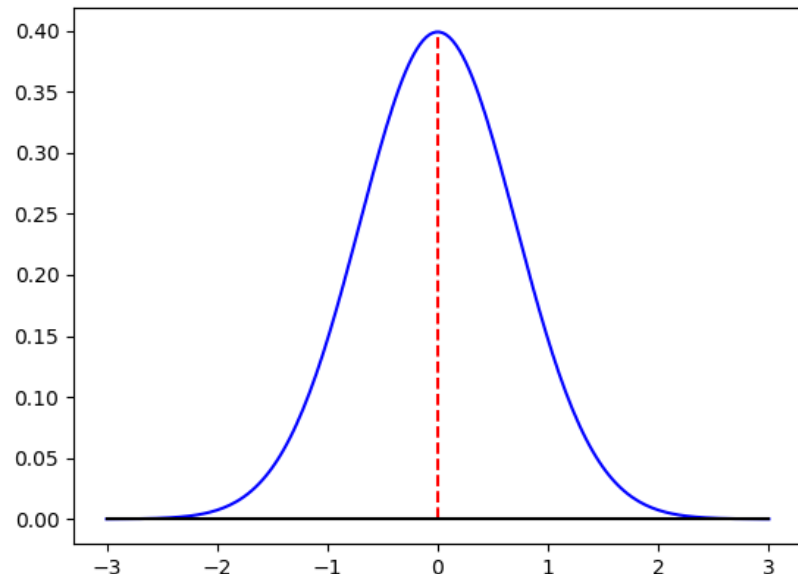
Filtro 11x11



IL FILTRO GAUSSIANO (II)

- Il filtro box è un rudimentale filtro per la sfocatura e la riduzione del rumore o del dettaglio
- La riduzione del rumore è fondamentale specialmente nelle immagini vecchie o rovinate
- Il problema con il filtro box è che tutti i pixel interessati vengono pesati in maniera uguale

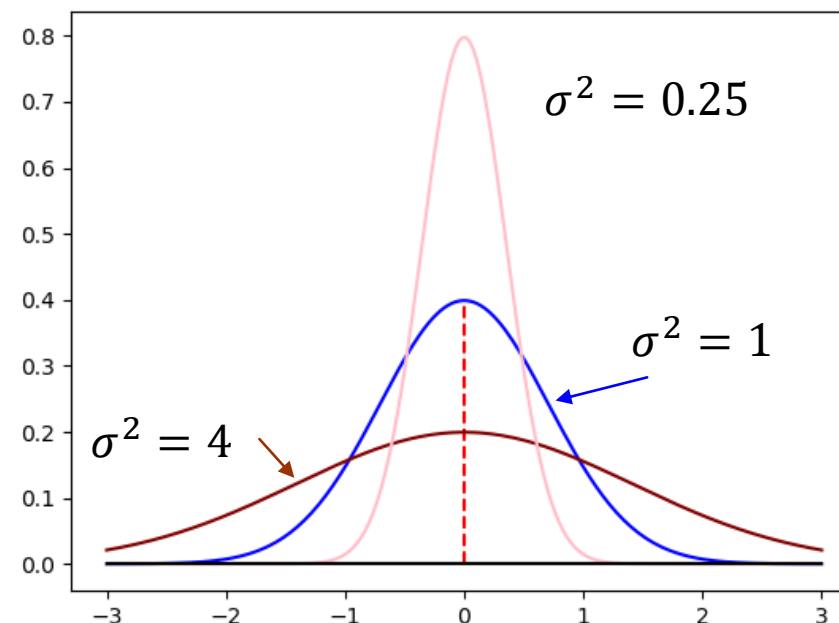
IL FILTRO GAUSSIANO (II)



$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{\sigma^2}}$$

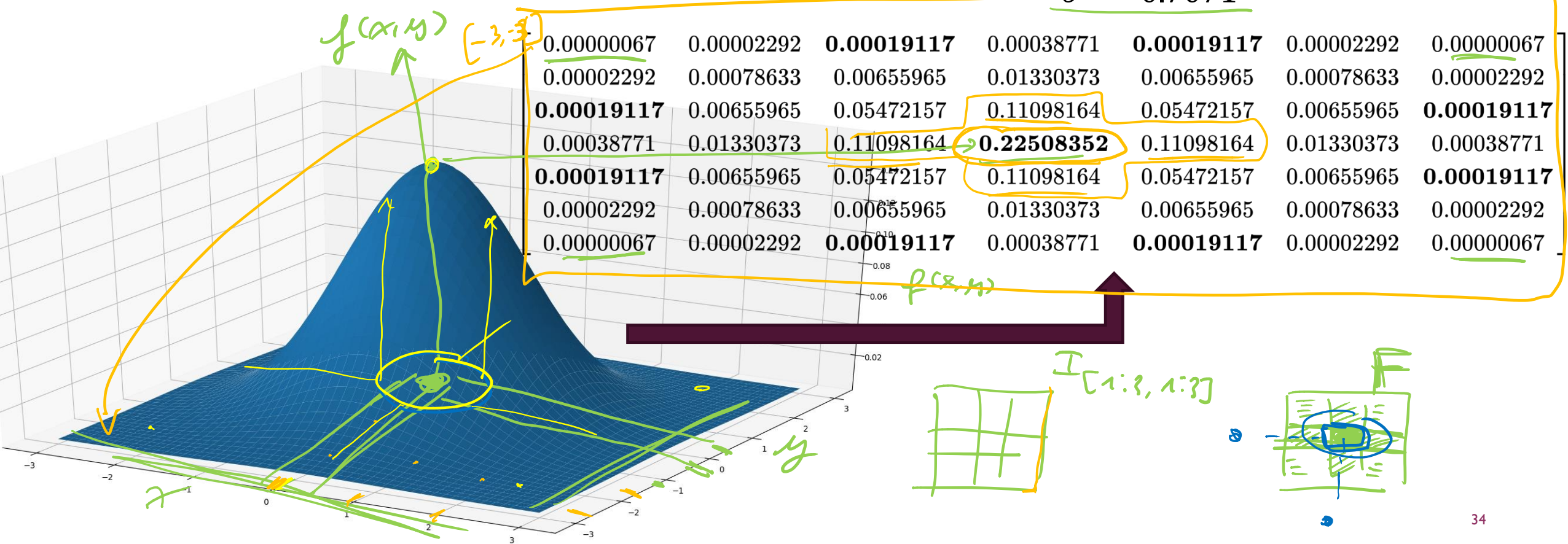
Il parametro σ^2 («varianza») governa l'ampiezza della curva

La funzione gaussiana (o normale) standard è una funzione «a campana» in cui lo zero ha un valore molto elevato, e quest'ultimo decresce dolcemente fino a quasi assestarsi verso lo zero.



IL FILTRO GAUSSIANO (III)

Il filtro gaussiano può essere esteso alle due dimensioni in maniera molto intuitiva



IL FILTRO GAUSSIANO (IV)

$$\sigma^2 = 1$$



$$\sigma^2 = 4$$



IL FILTRO GAUSSIANO (IV)

In linea di massima, funziona anche con un'immagine a colori



$$\sigma^2 = 4$$



FILTRO MEDIANO (I)

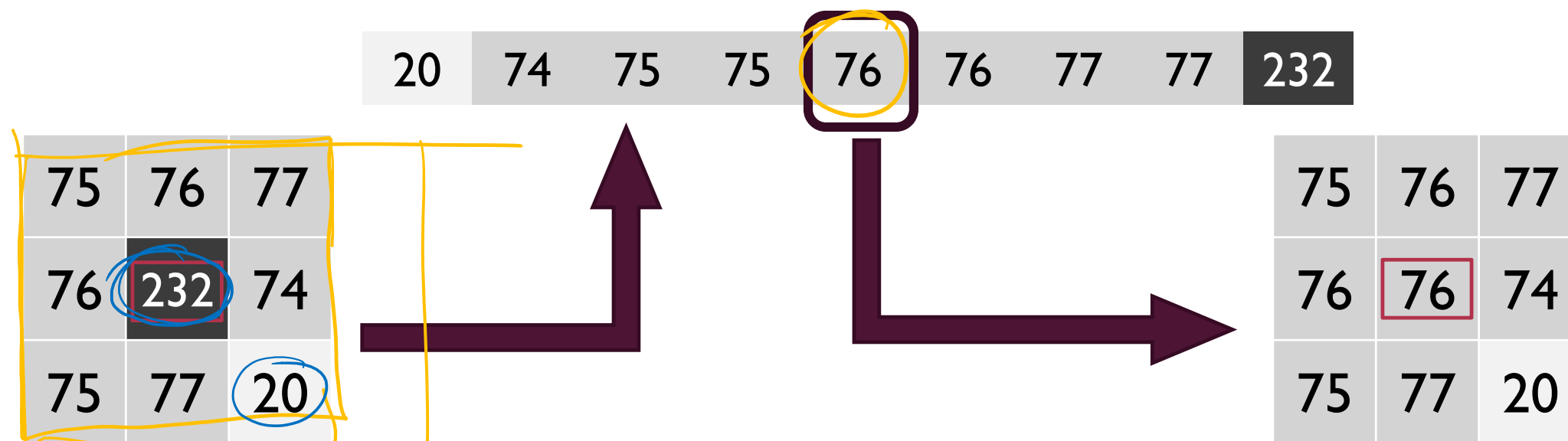
- Mediana (chi sa cos'è?)
- dato un insieme di n punti ordinati (n dispari), la mediana è il punto che si trova in posizione $\left\lceil \frac{n}{2} \right\rceil$ → ~~Algoritmo~~ ALGORITMO PER ECCELLENZA $12/2 = 6$



FILTRO MEDIANO (II)

NON PARAMETRICO

- Il filtro mediano sostituisce il valore mediano dei pixel all'interno della finestra



Il filtro mediano fa **prevalere la maggioranza sul singolo.**

FILTRO MEDIANO (III)

- Data la capacità della mediana di **isolare i valori eccezionali («outlier»)** all'interno di un insieme di punti, è particolarmente indicato per eliminare piccoli disturbi o impurità dalle immagini
- Il suo effetto risulta essere particolarmente più *morbido* rispetto ad un filtro medio

original



added noise



average ✕



median ✕



EDGE DETECTION

- Traducibile con «**riconoscimento dei bordi**»
- Un bordo è un segmento/un arco in cui vi è un **repentino cambio di intensità**

Hand-drawn 4x4 grid illustrating edge detection values:

0	0	25	25
0	0	25	25
0	0	25	25
0	0	25	25



Il bordo rappresenta uno degli esempi più basilari di ***feature di basso livello***

FILTRO DI SOBEL

■ Filtraggio in due passaggi



★

-1	-2	-1
0	0	0
1	2	1

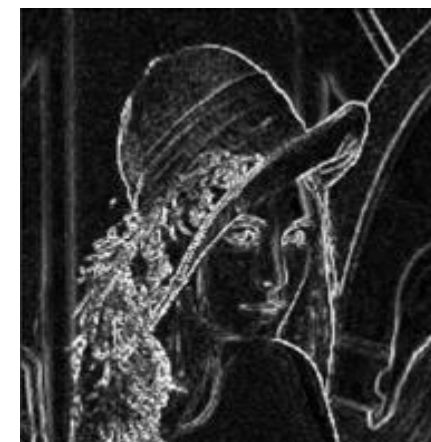
 $= G_y$

-1	0	1
-2	0	2
-1	0	1

★

 $= G_x$

$$\sqrt{G_x^2 + G_y^2} =$$



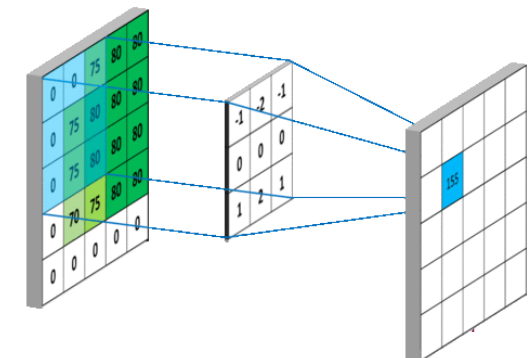
Nota: a volte un eccesso di dettaglio (es. risoluzione troppo alta) può causare un **eccesso di bordi** come risposta del filtro di Sobel. Usualmente è necessario **filtrare preventivamente l'immagine con un filtro gaussiano** per ridurre il dettaglio.

EDGE DETECTION (II)

- Nel frattempo, sono stati sviluppate altre varianti di edge detection molto più performanti, non trattate per complessità
- Es. Canny Edge Detector (1986)

RIASSUMENDO

- «Feature» può essere tradotto come «caratteristica» di un'immagine
- È una **parte d'informazione** dell'immagine **utile al conseguimento di un determinato compito**
- Feature detection = Riconoscimento di caratteristiche
- Compito difficile
- Storicamente conseguito grazie all'applicazione di filtri tramite correlazione
- Correlazione = sostituzione di ogni pixel tramite una «media» dei pixel vicini
- Si può pensare come una «finestrella» che spazza l'immagine pixel per pixel
- I filtri possono avere vari effetti: riduzione del dettaglio, evidenziazione dei bordi, rimozione rumore...

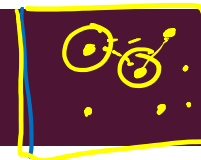


FEATURE DETECTION (APPLICAZIONE)

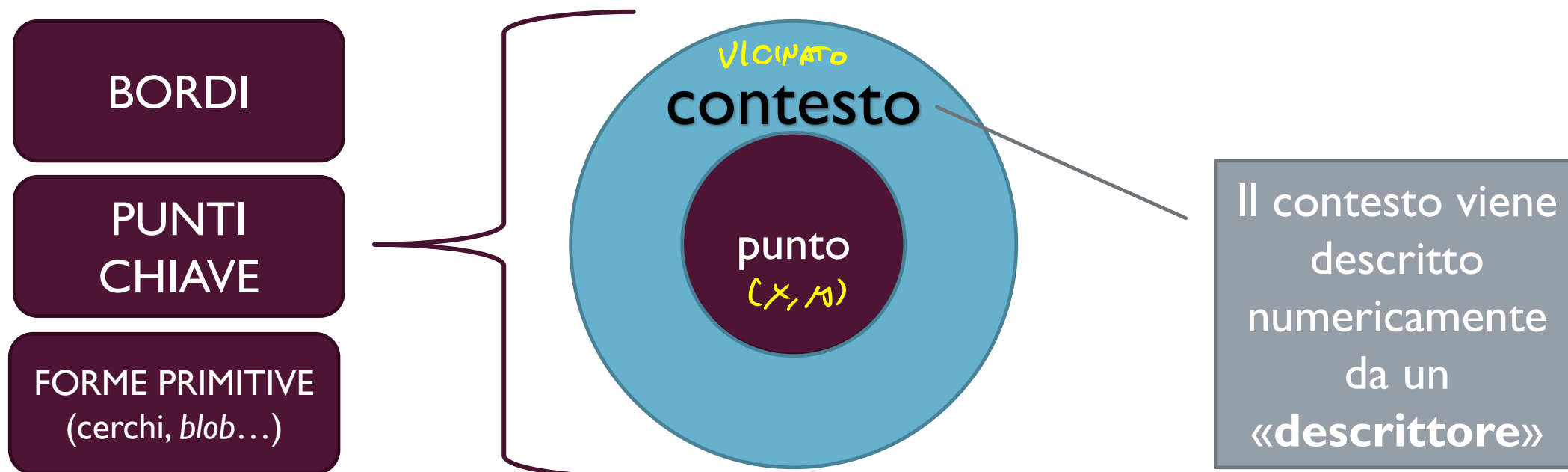
- Obiettivo ideale: riconoscere oggetti simili in immagini diverse
 - Differente orientamento
 - Differente illuminazione
 - Differente contesto
 - ...

«Invarianza»

IL PUNTO CHIAVE

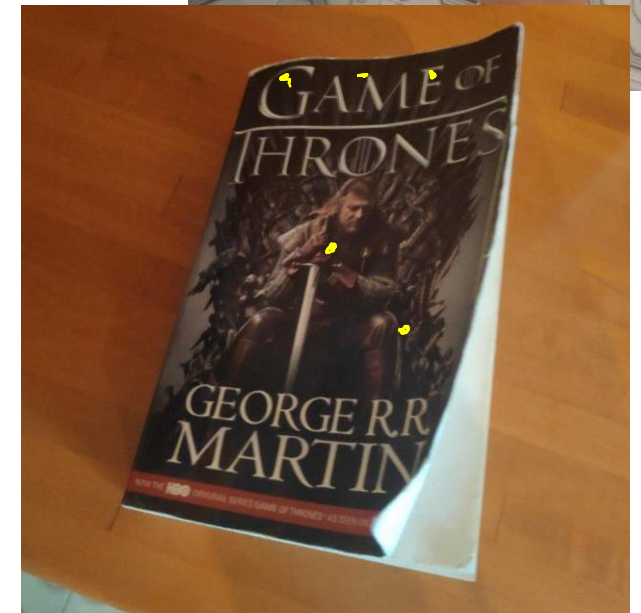
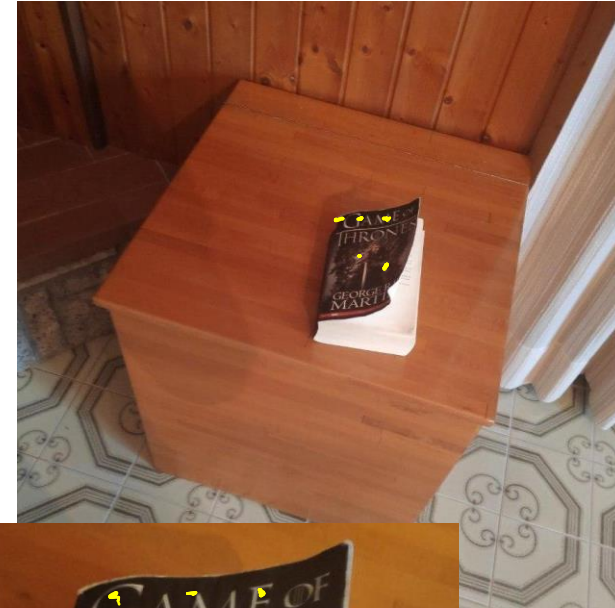


- Per permettere il riconoscimento di **oggetti complessi**, operiamo una **decomposizione in caratteristiche (feature) semplici**



SIFT (I)

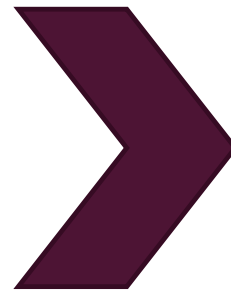
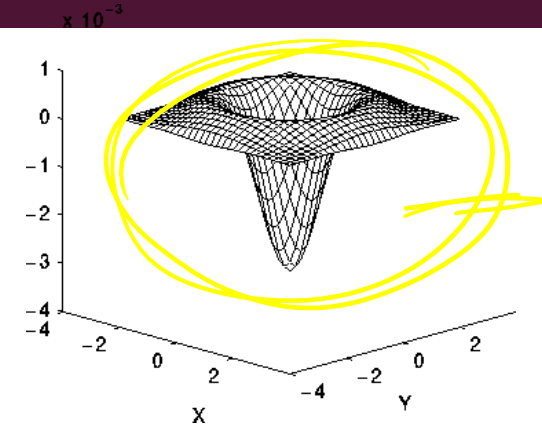
- Acronimo di **Scale Invariant Feature Transform**
- **Scale Invariant** = *Invarianza alla scalatura*
- (ES. Sobel → può dare risultati non univoci in base alla risoluzione o scala di un oggetto)
- SIFT si rende **invariante** alla scala:
- Indipendentemente dalla grandezza dell'oggetto nell'immagine e dalla risoluzione di questa, i punti chiave dell'oggetto identificati da SIFT saranno sempre gli stessi



SIFT (II)

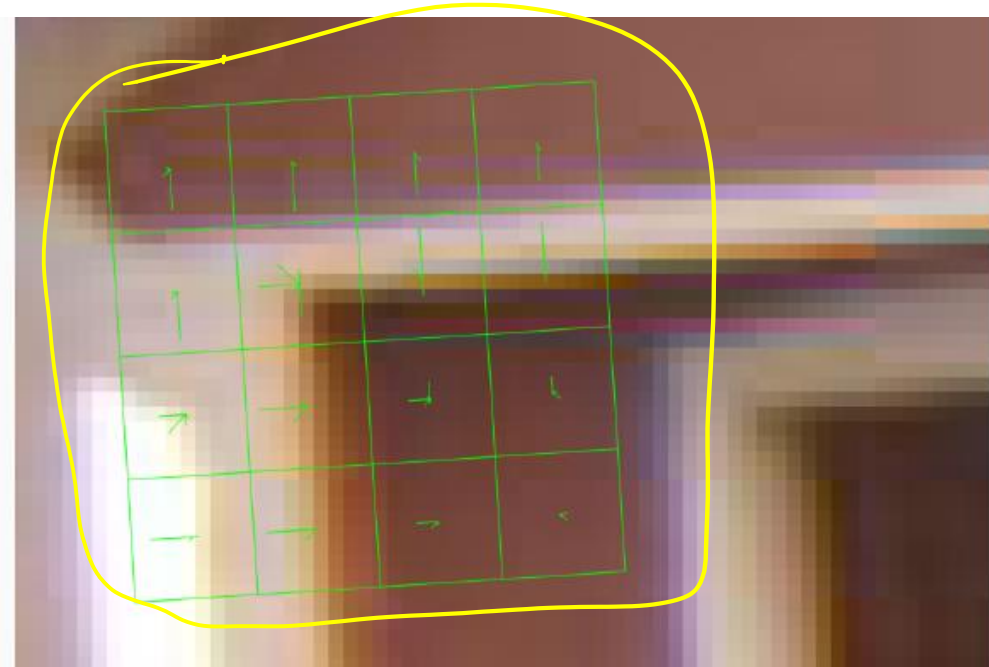
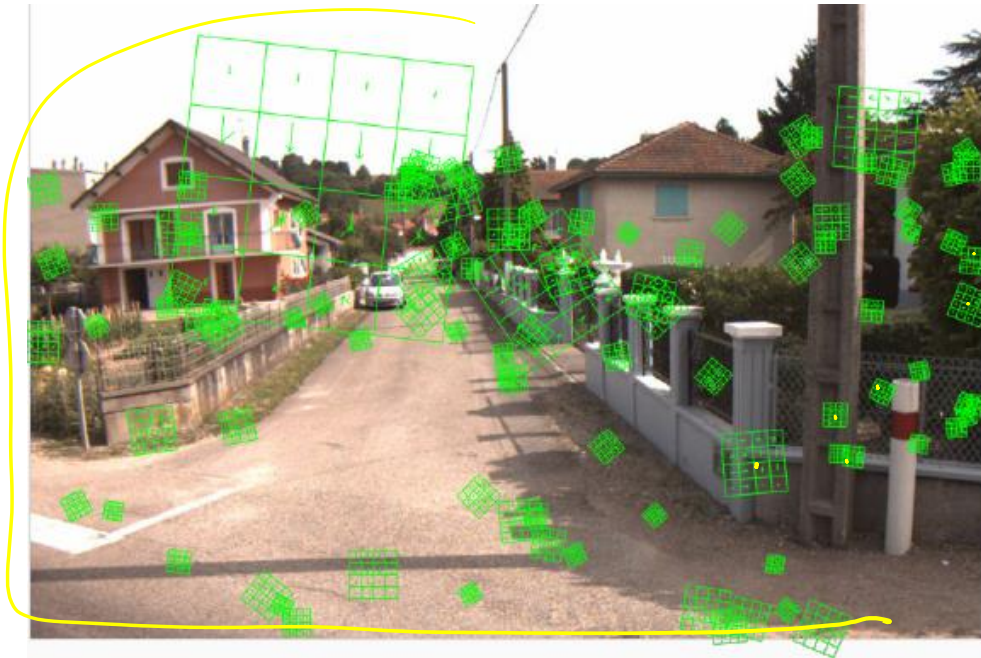


Basato sul **riconoscimento dei «blob»**



SIFT (II)

- Al centro del blob viene aggiunto un **descrittore numerico** del vicinato del punto



APPLICAZIONE DI SIFT (I)

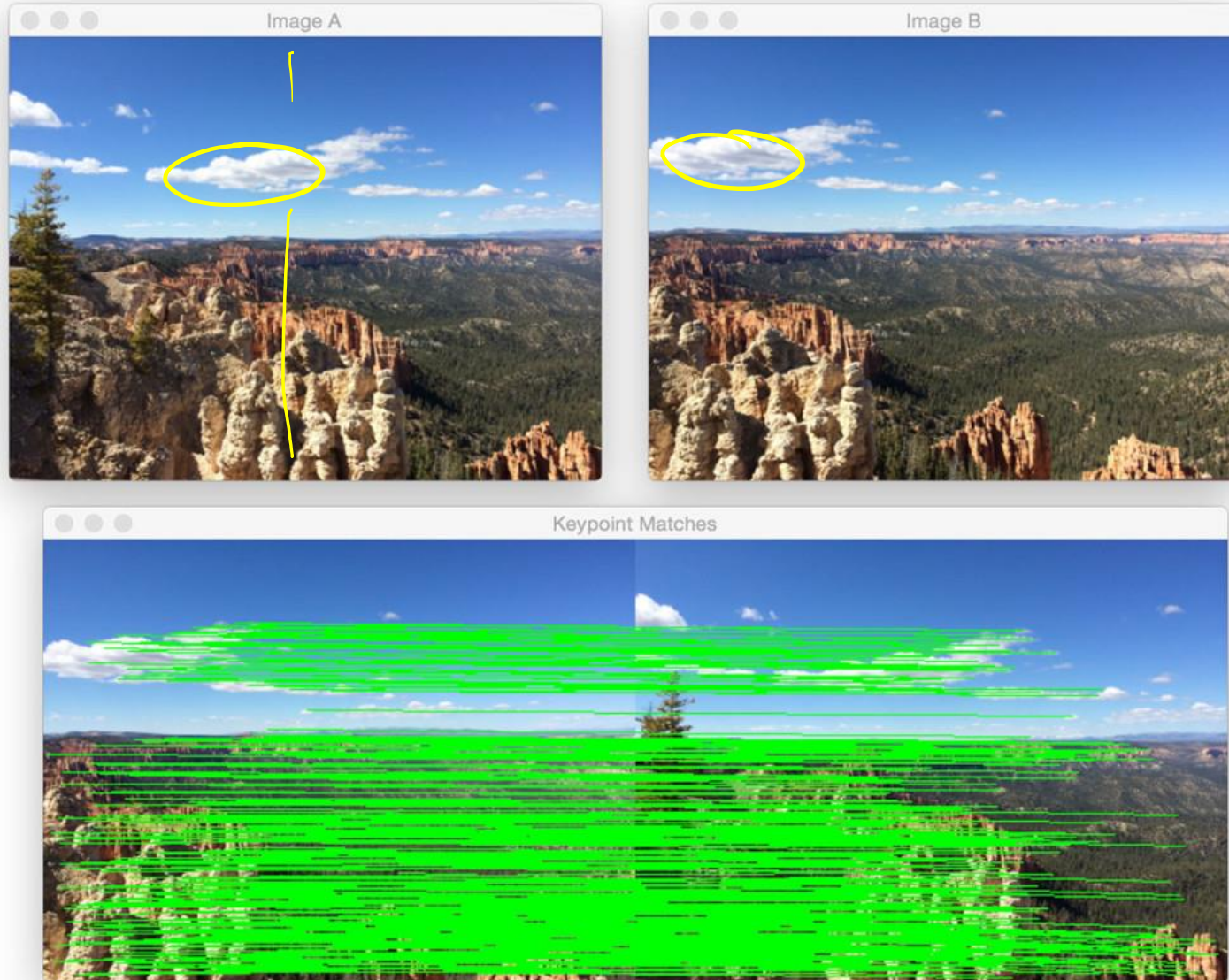
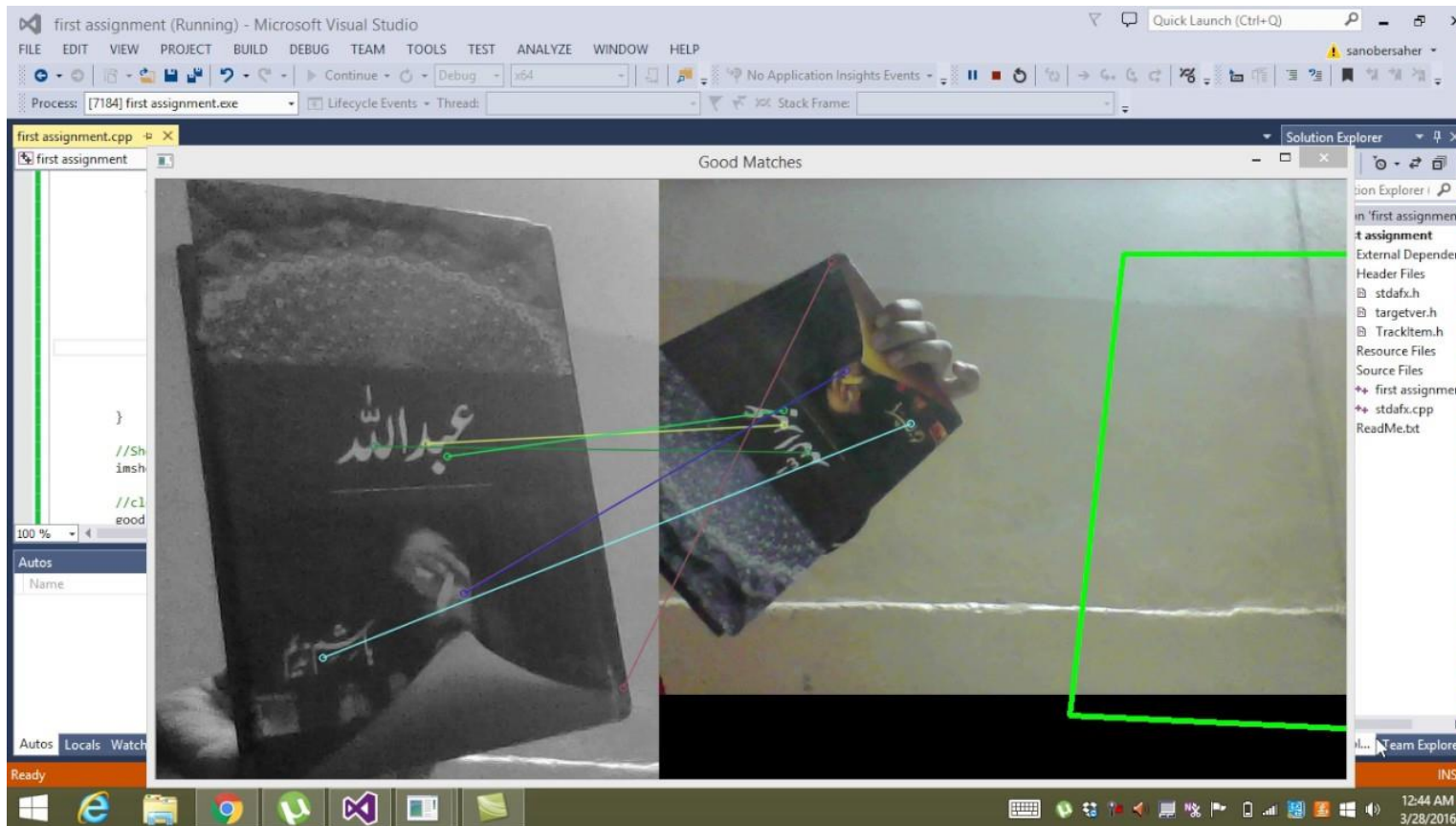


IMAGE STITCHING

Source:

https://www.pyimagesearch.com/wp-content/uploads/2016/01/bryce_match_01-1024x814.jpg

APPLICAZIONE DI SIFT (II)

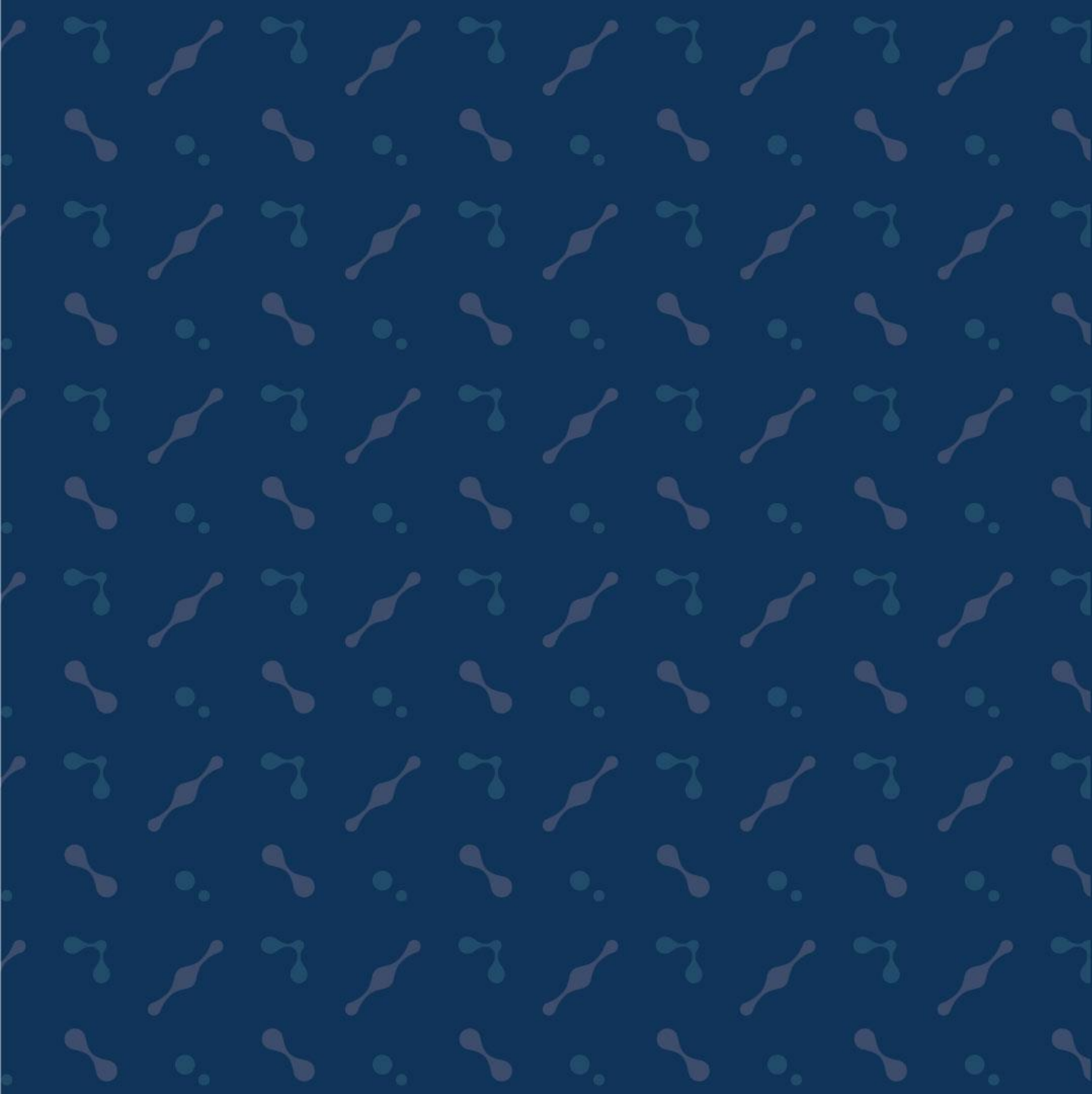


- Object tracking («Tracciamento di oggetti»)

Source: https://i.ytimg.com/vi/q7_BhXeWv6I/maxresdefault.jpg

RIASSUMENDO

- Partendo dai filtri, è possibile progettare in maniera intelligente degli algoritmi per il riconoscimento di caratteristiche
- Idealmente, vorremmo che un oggetto venga riconosciuto indipendentemente dalle condizioni di illuminazione, dall'orientamento, dalla vicinanza/lontananza dall'obiettivo
 - Concetto di **invarianza**
- Si può riconoscere un oggetto identificando **punti chiave** di quest'ultimo
- SIFT identifica punti chiave basandosi sui blob (forme «approssimativamente circolari»)
- E ne descrive il «vicinato» tramite un descrittore numerico
- Il modo in cui SIFT è progettato, permette di identificare i pt. chiave indipendentemente dalla grandezza dell'oggetto



.03

**Dal modello lineare
alla rete neurale**

IL DATASET

Unità (statistiche)
Osservazioni

Variabili

Unità	Altezza (cm)	Peso (kg)	Età (anni)	Sesso
1	175	70	21	M
2	167	58	24	F
3	182	72	22	M
4	177	81	45	M
5	174	64	30	F
6	162	53	37	F
...				
<i>n</i>	178	60	19	F

RELAZIONE FRA VARIABILI

- Potrei chiedermi se esiste una legge che governa la relazione fra due o più variabili
- Es: *il peso e l'altezza sono in qualche modo collegati?*
- → *Posso in qualche modo prevedere il peso data l'altezza?*
- *Con che sicurezza / precisione posso formulare la previsione?*

RELAZIONE LINEARE

- Il collegamento più semplice a cui posso pensare è la **relazione lineare** fra le due variabili
- *Il peso è determinato dall'altezza, moltiplicata per un determinato **valore fisso**, più un eventuale **ammontare fisso** indipendente dall'altezza*

Coefficiente
angolare

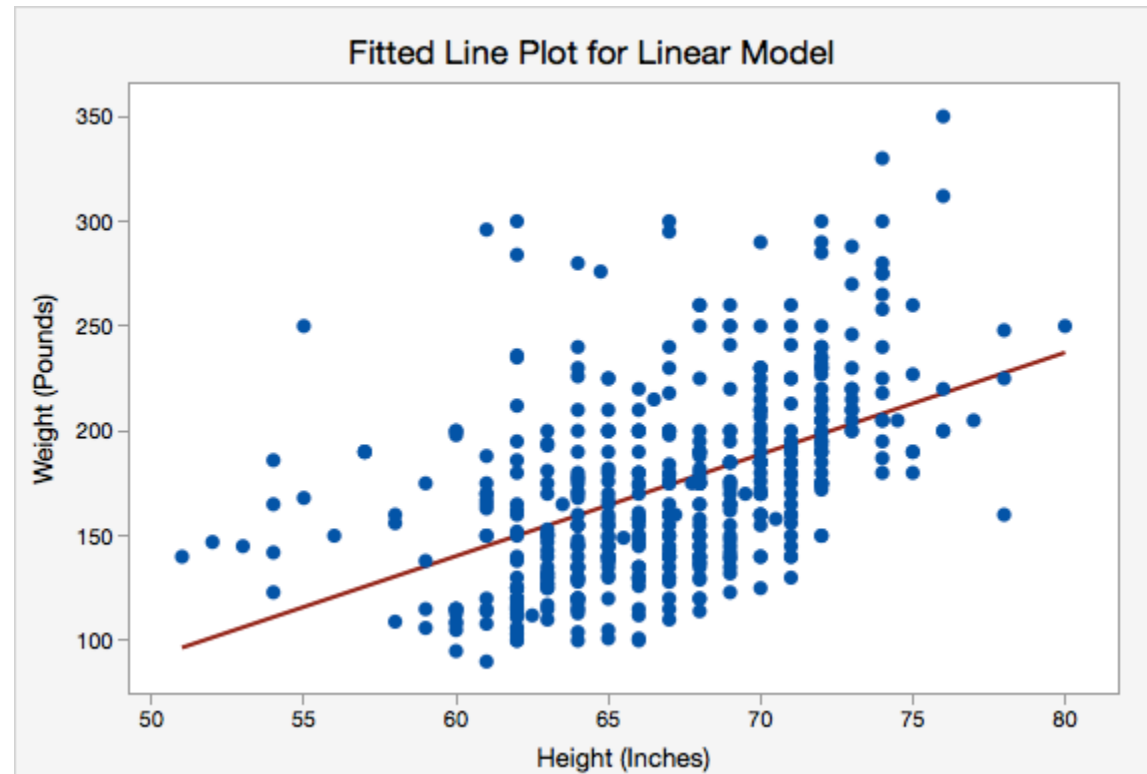
Intercetta

$$\text{Peso} = m \cdot \text{altezza} + q$$

Responso

Covariata/e

RELAZIONE LINEARE (GRAFICO)



LA FUNZIONE DI PERDITA / ERRORE / COSTO

- Devo scegliere un criterio per determinare la retta
- Se i punti non sono allineati, andrò sempre incontro ad un errore scegliendo una retta piuttosto che un'altra
- Idea: voglio **minimizzare** questo errore
- E voglio che gli errori più *gravi* vengano penalizzati più *gravemente*

L'ERRORE QUADRATICO

- **Errore quadratico**
- Considero la differenza fra il valore previsto dalla retta e il valore osservato reale

PEso *PEso PREVISTO = m · at + q*

↓ ↓

$$EQ(y, \hat{y}) = (y - \hat{y})^2$$

Domanda: perché quadratico? Perché non semplicemente $(y - \hat{y})$?

L'ERRORE QUADRATICO MEDIO

- (Ricorda) normalmente abbiamo n unità statistiche, non una sola
- Possiamo ottenere l'errore quadratico per n unità semplicemente sommando questo errore per ognuna delle osservazioni

$$\mathcal{L}(y, \hat{y}) = (y_1 - \hat{y}_1)^2 + \dots + (y_n - \hat{y}_n)^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Problema: calcoliamo la perdita per un campione di 2 unità. Ora ripetiamo l'esperimento su 5 unità. Notate qualcosa di strano?

$$\mathcal{L}(y, \hat{y}) = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}$$

Errore
quadratico medio

MOMENTO INTERATTIVO – ALTEZZA VS NR. SCARPE

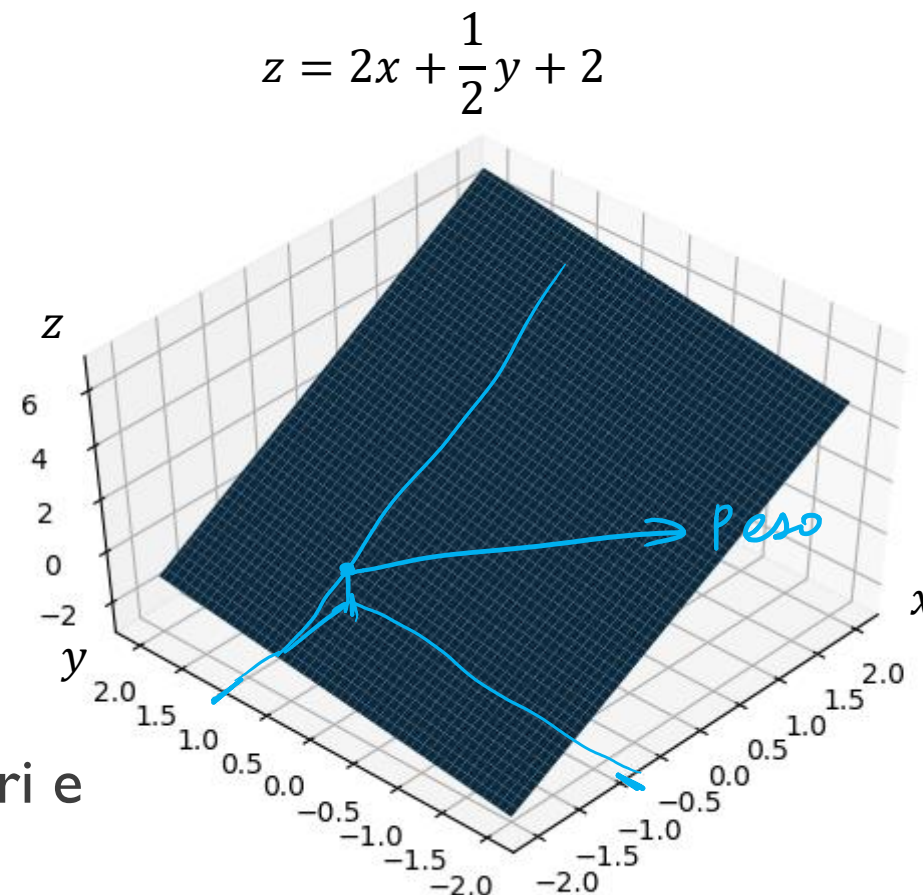
RELAZIONE LINEARE A PIÙ VARIABILI

- Posso aggiungere ulteriori variabili per determinare il peso di una persona

$$z = m_1 x + m_2 y + q$$

- $Peso = m_1 \cdot altezza + m_2 \cdot età + q$

- Il significato geometrico non cambia
- Ora ho tre dimensioni (altezza, età, peso)
- La retta in 2D equivale ad un piano in 3D
 - Il piano è determinato dai due coefficienti angolari e dalla quota

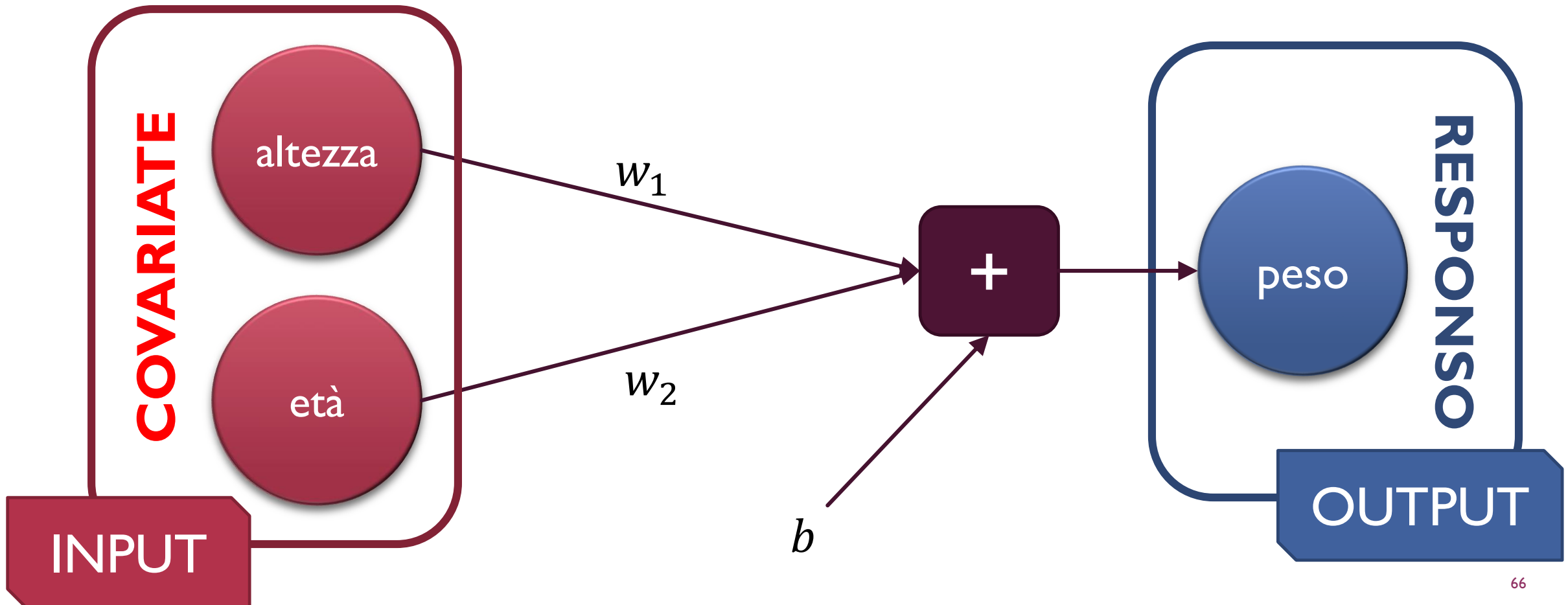


NOZIONI E NOMENCLATURA

- Modello lineare generico: $y = m_1 \cdot x_1 + \dots + m_p \cdot x_p + q$
- In statistica, usualmente non si utilizzano i simboli m e q per indicare i coefficienti angolari e l'intercetta della retta
- Per i coefficienti di pendenza delle singole covariate, si usa β_i o w_i , per la quota β_0 o b .
- $y = b + w_1 \cdot x_1 + \dots + w_p \cdot x_p$
- La somma $w_1 \cdot x_1 + \dots + w_p \cdot x_p$ viene chiamata **somma pesata** o **combinazione lineare** di $x_1, \dots, x_p$ e i coefficienti $w_1, \dots, w_p$ sono detti **pesi** o **parametri**
- La quota b la chiameremo **bias**

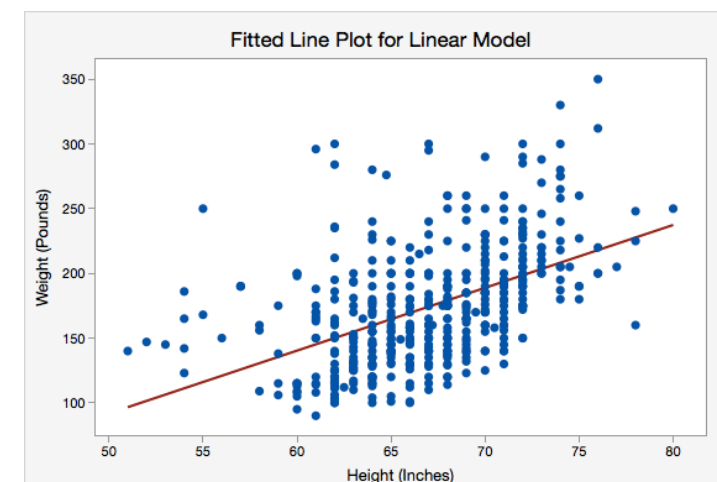
GRAFICO COMPUTAZIONALE DEL MODELLO

$$peso = b + w_1 \cdot altezza + w_2 \cdot eta$$



RIASSUMENDO

- Voglio studiare la variazione di un fenomeno in dipendenza di una variabile
- Es. Peso in relazione ad altezza
- Posso pensare ad una relazione lineare:
- $Peso = m \cdot altezza + q$
- La retta viene costruita in modo tale da “passare in maniera ottimale” attraverso i vari punti
- La retta “ottima” minimizza una funzione $\mathcal{L}$ detta errore o Perdita
- $\mathcal{L}$ aumenta all’aumentare della “distanza” fra retta e punti
- $y = w_1x_1 + \dots + w_px_p + b$

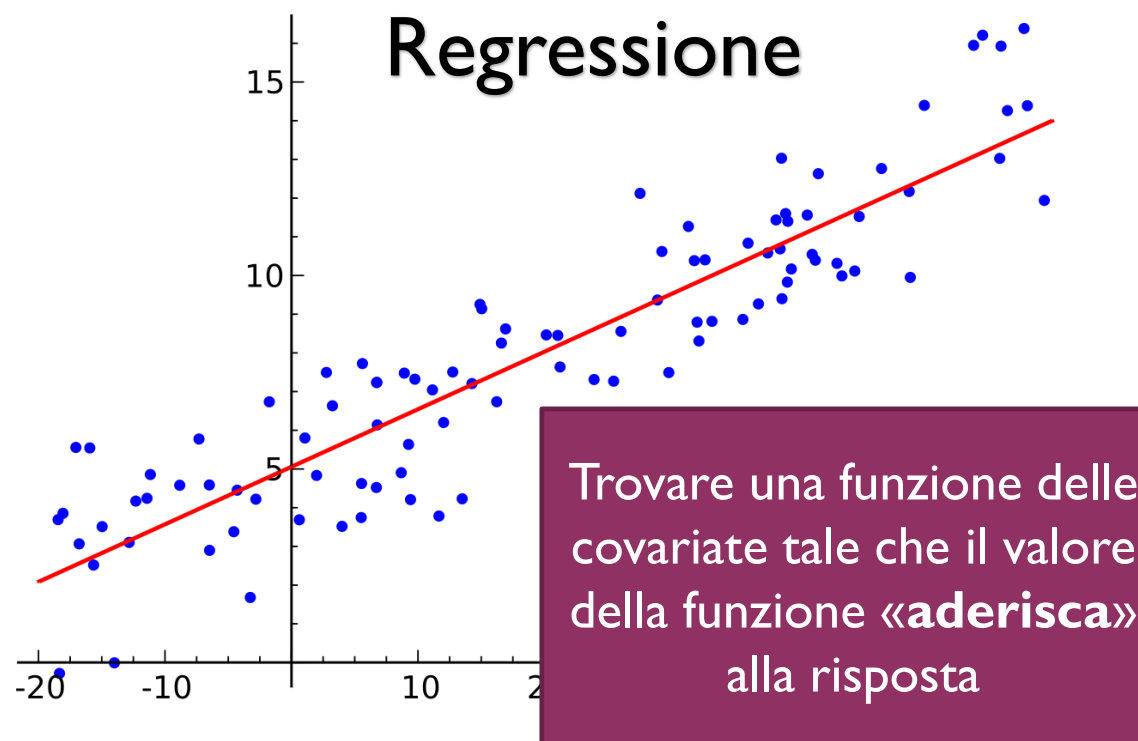


CLASSIFICAZIONE

- Finora abbiamo visto casi in cui il responso è un numero reale
 - REGRESSIONE
- Potremmo avere casi in cui il responso è una categoria
 - Es. determinare se in un'immagine è presente un GATTO o un CANE
 - CLASSIFICAZIONE

REGRESSIONE VS. CLASSIFICAZIONE

Trovare una funzione che metta in relazione la risposta e le covariate



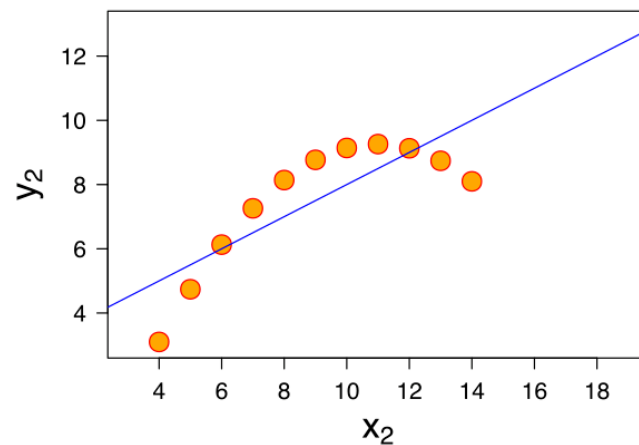
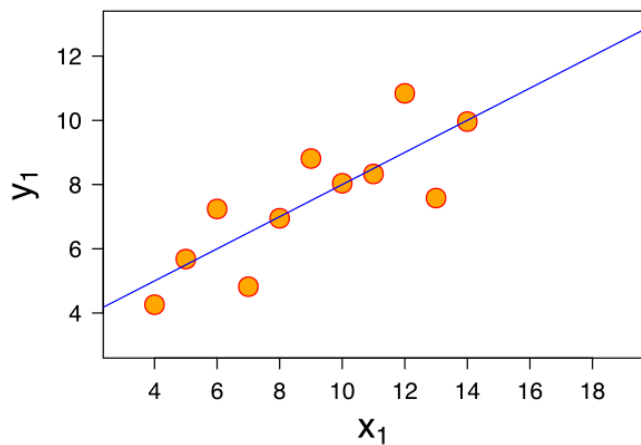
Classificazione

Trovare una funzione delle covariate tale che il valore della funzione «separi» le categorie della risposta

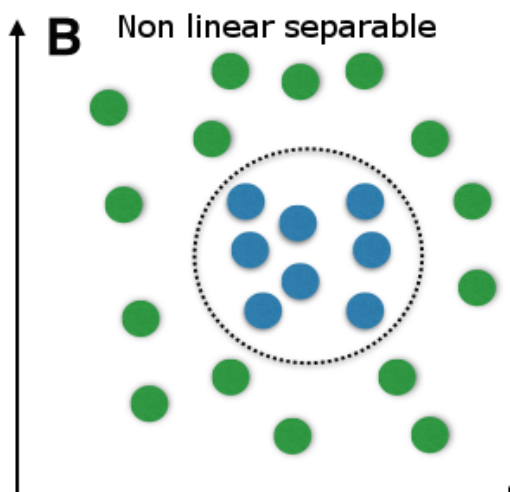
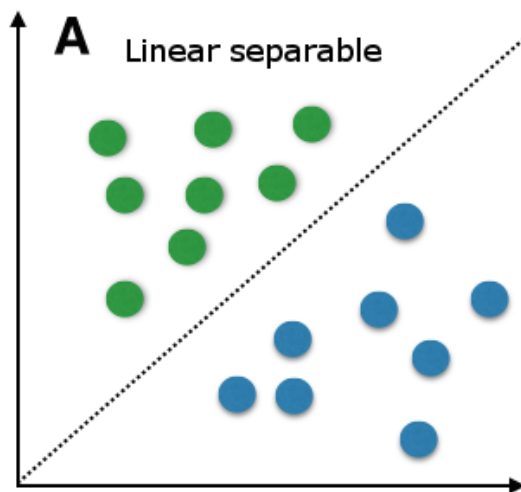
RELAZIONI NON-LINEARI

- I modelli lineari sono fra i modelli più studiati della statistica e del *machine learning*
- I modelli lineari hanno garanzie teoriche sulla precisione e sull'affidabilità dei propri risultati
- Problema: una grandissima parte delle relazioni fra fenomeni del mondo reale è altamente **non-lineare**
- ... e in questi fenomeni è coinvolto un grandissimo numero di variabili

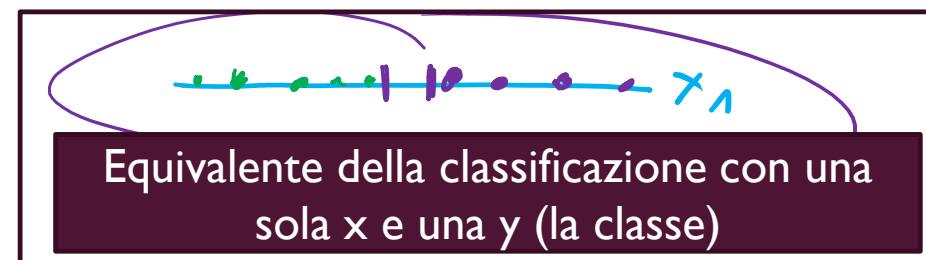
RELAZIONI NONLINEARI - GRAFICO



Regressione



Classificazione

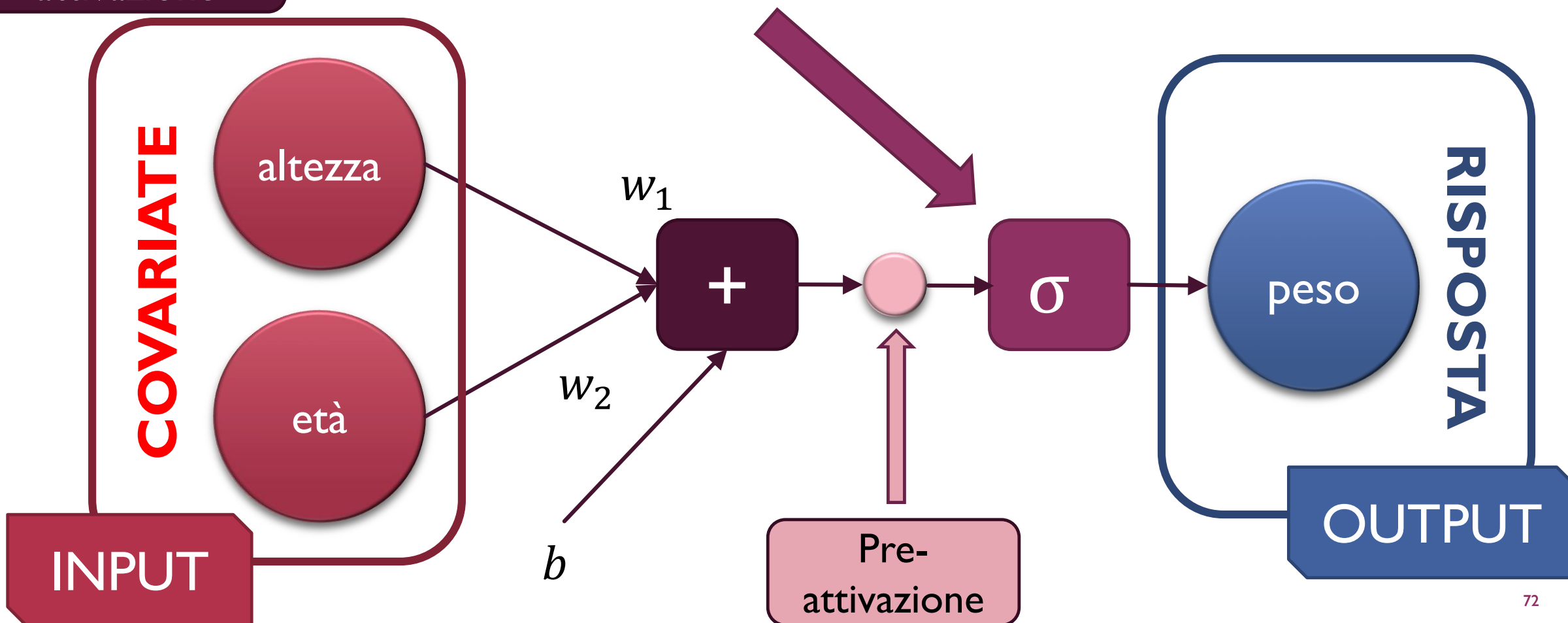


MODELLO NON-LINEARE (ESEMPIO)

Funzione di
attivazione

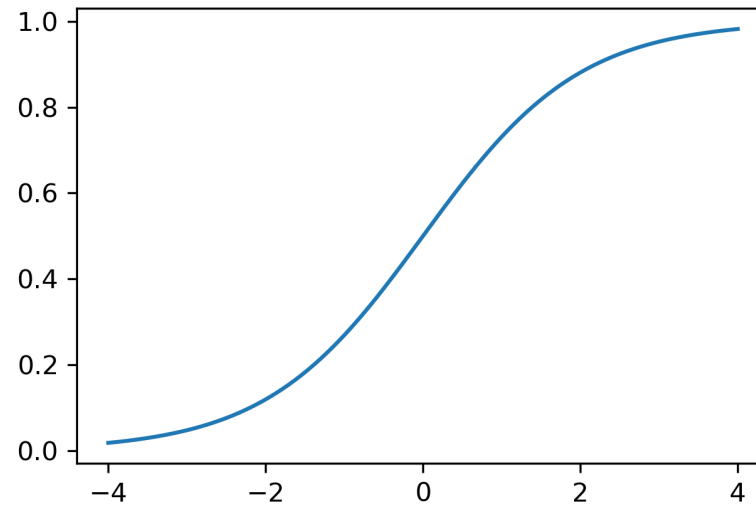
$$\sigma: \mathbb{R} \rightarrow \mathbb{R}$$

$$peso = \sigma(b + w_1 \cdot altezza + w_2 \cdot eta)$$

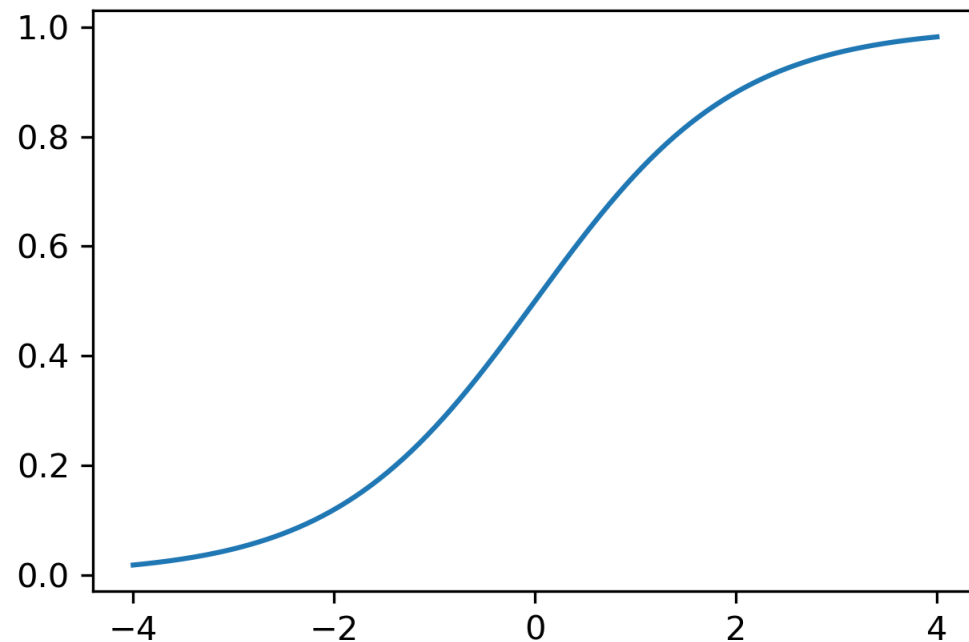


LA FUNZIONE DI ATTIVAZIONE

- La funzione di attivazione introduce la non-linearità fra covariate e risposta
- Aggiunge espressività al modello
 - = il modello può esprimere più espressioni di relazione fra x e y



ESPRESSIONE DELLA FUNZIONE SIGMOIDE



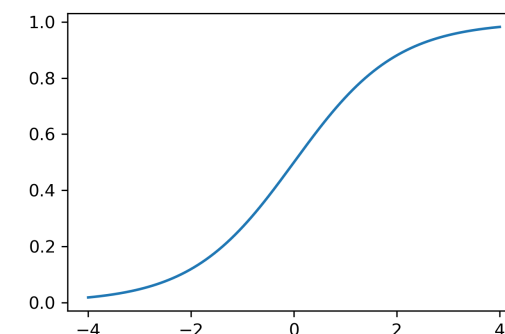
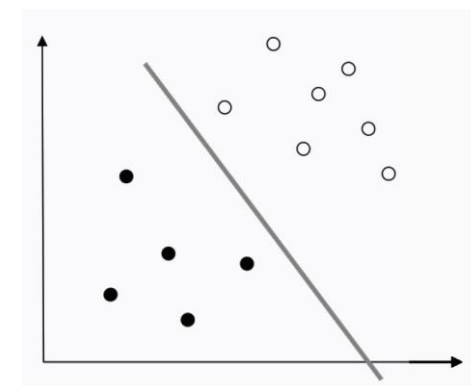
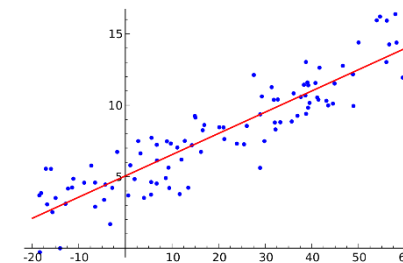
Massimo = 1

$$y = \sigma(w_1x_1 + \dots + w_px_p + b) = \sigma(a) \in [0,1]$$

Minimo = 0

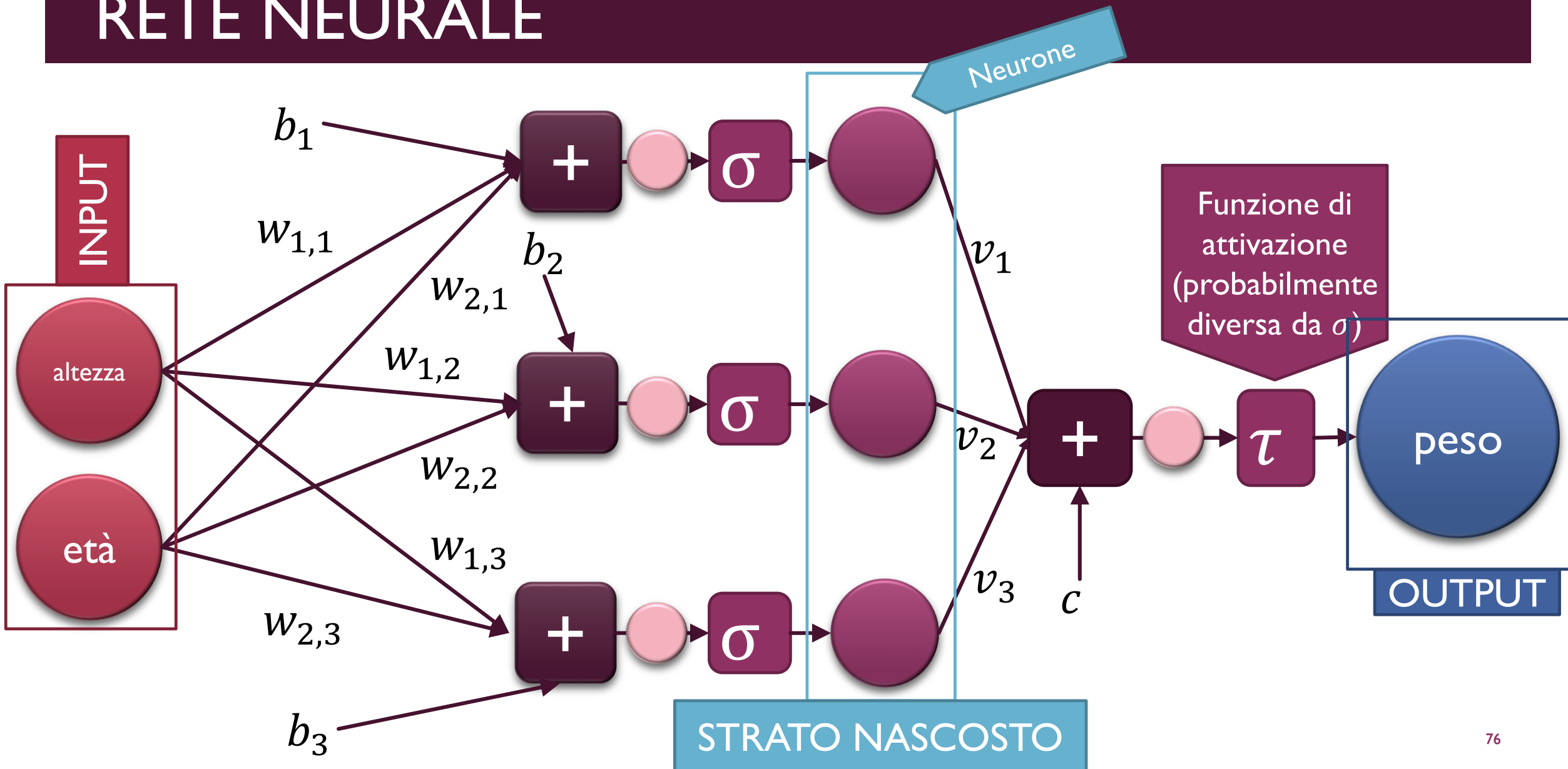
RIASSUMENDO

- Se la risposta assume valori continui, ho un problema di regressione (devo determinare la retta che «passa meglio» fra i punti)
- Se la risposta è di tipo categorico, ho un problema di classificazione (devo determinare la retta che «divide meglio» i punti)
- Es. di variabile di tipo categorico: Malattia SÌ / NO; Disturbo LIEVE / MEDIO / GRAVE
- Gran parte delle relazioni naturali è di tipo NON LINEARE
- È possibile modellare la relazione lineare in non-lineare aggiungendo una funzione non-lineare all'equazione della retta:
- $y = \sigma(w_1x_1 + \dots + w_px_p + b)$



σ :

RETE NEURALE



LO STRATO NASCOSTO

$$y = \sigma\left(\sum_{i=1}^p w_i x_i + b\right)$$

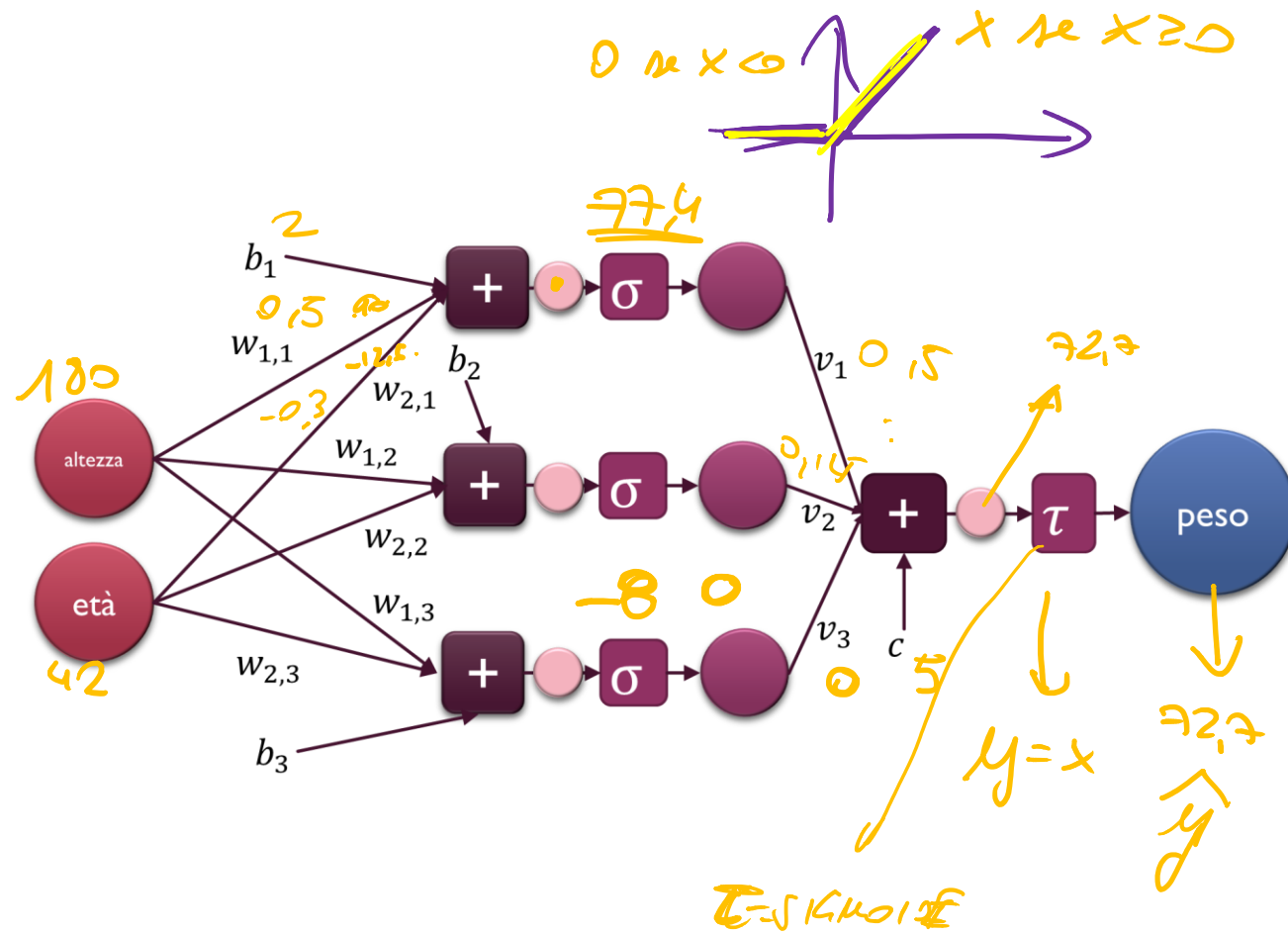
- L'aggiunta dello strato nascosto è la chiave del successo delle reti neurali rispetto a altri modelli di regressione o classificazione lineari e non-lineari
- Il modello diventa un APPROSSIMANTE UNIVERSALE = può approssimare (con un numero sufficientemente alto di neuroni e un determinato livello di *tolleranza*) **qualsiasi tipo di funzione esistente.**

ESERCIZIO

- Prendiamo il caso della rete neurale descritto nella slide precedente
- Supponiamo di avere un'unità statistica con le seguenti osservazioni:
 - altezza = 180 cm, età = 42 anni
- Supponiamo ora che i valori dei pesi e dei bias siano i seguenti:

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \end{bmatrix} = \begin{bmatrix} 0,5 & 1,0 & 0,2 \\ -0,3 & 1,0 & -1,0 \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 2,0 \\ 2,0 \\ -2,0 \end{bmatrix}$$

- Calcolare i valori delle tre pre-attivazioni dei neuroni nascosti
- Considerata la funzione di attivazione $RELU(x) = \begin{cases} x, & \text{se } x \geq 0 \\ 0, & \text{altrimenti} \end{cases}$ calcolare anche i valori delle attivazioni
- Infine, dati $v = \begin{bmatrix} 0,5 \\ 0,125 \\ 4,1 \end{bmatrix}$, $c = 5$, $\tau(x) = x$, si calcoli $\hat{y}$, il valore di peso previsto dal modello



RAFFIGURAZIONE PIÙ PROFESSIONALE

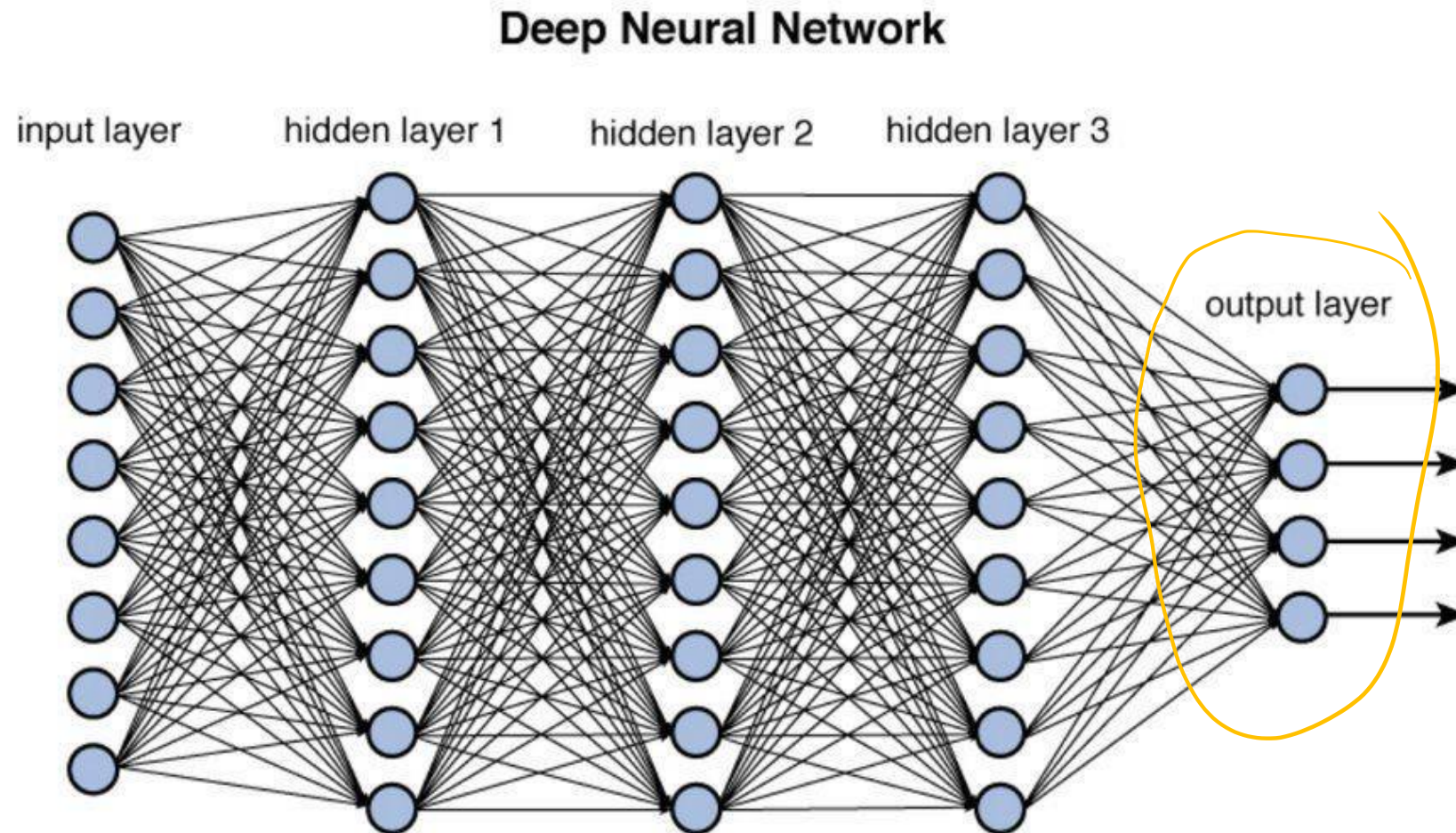


Figure 12.2 Deep network architecture with multiple layers.

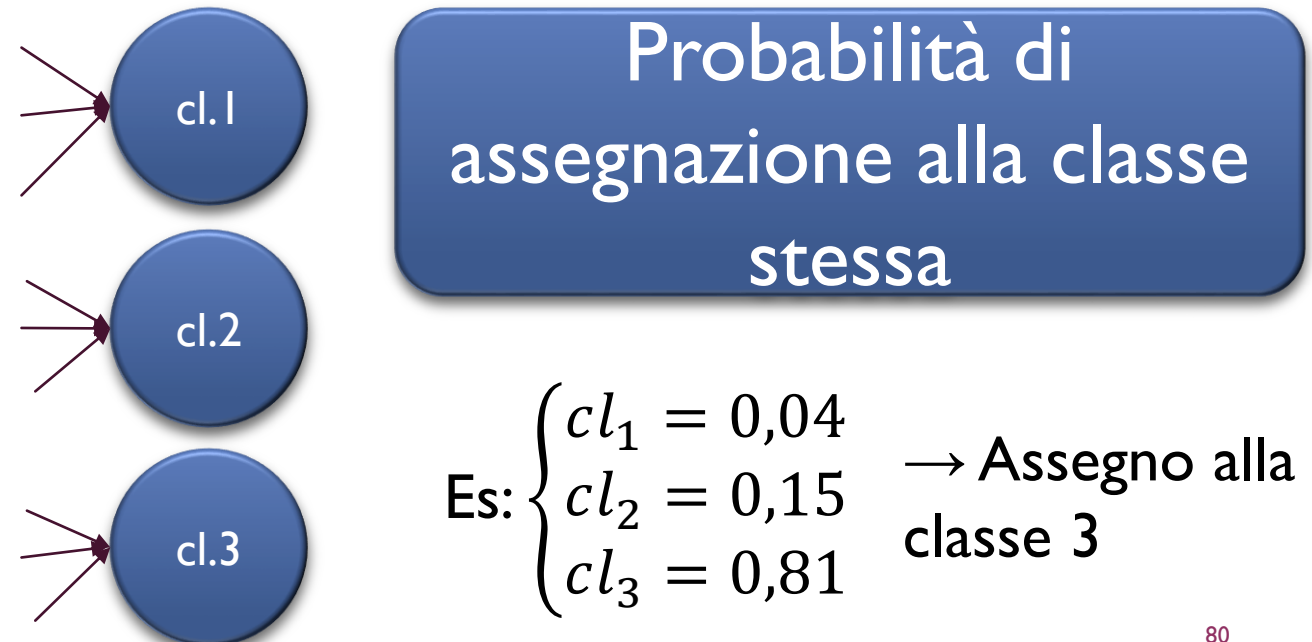
RETE PER REGRESSIONE VS. CLASSIFICAZIONE

- Cambia solamente lo strato di output

REGRESSIONE



CLASSIFICAZIONE



CLASSIFICAZIONE DI IMMAGINI

- Le reti in assoluto più comuni sono quelle progettate per la classificazione di immagini

CANI



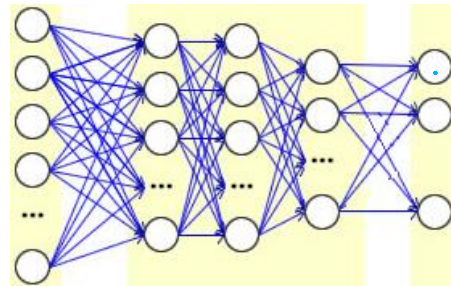
GATTI



PESCI



NUOVA
IMMAGINE



$$\begin{cases} \text{cane} = 0,12 \\ \text{gatto} = 0,86 \\ \text{pesce} = 0,02 \end{cases}$$



È un
gatto!

RIASSUMENDO

- Espando la relazione non-lineare precedente
- Inframmezzando degli strati intermedi detti **strati nascosti**
- Ogni strato intermedio (e finale) ha la sua relazione non-lineare con la sua funzione di attivazione
- Il numero di strati intermedi è arbitrario
- Lo strato finale ha:
 - 1 neurone in caso di regressione
 - c neuroni in caso di classificazione ($c = \text{nr. Classi}$)

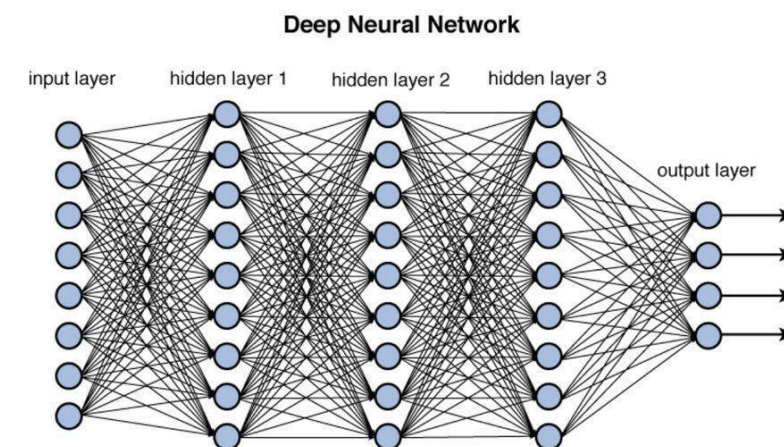
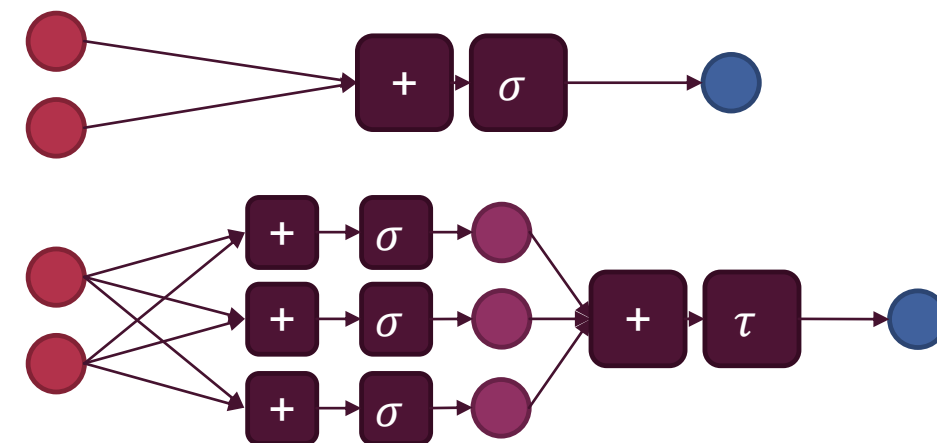


Figure 12.2 Deep network architecture with multiple layers.

VALUTAZIONE DELLE PERFORMANCE

- Abbiamo visto prima come un modello venga addestrato sulla base di una funzione $\mathcal{L}$ chiamata perdita o costo.
- Il problema con $\mathcal{L}$ è che, di solito, restituisce valori assoluti e non relativi
 - Es. valore assoluto: $l \in \mathbb{R}$ o $l \in \mathbb{R}^+$
 - Es. valore relativo: $r \in [0,1]$
- Un esempio di valore relativo per una rete di classificazione:
accuratezza

$$\text{Accuratezza} = \frac{\text{Unità classificate correttamente}}{n}$$

72 su 100 $\rightarrow 72\%$ $0,72$

SOVRADATTAMENTO (OVERFITTING)

- Se consideriamo le performance calcolate sulla base delle unità di addestramento, c'è un problema
- Tenderemo a preferire modelli che si adattano completamente ai dati a disposizione (**overfitting**)
- Bisogna considerare però che il nostro modello deve **generalizzare** in modo da funzionare correttamente anche su nuovi dati, non visti in fase di addestramento



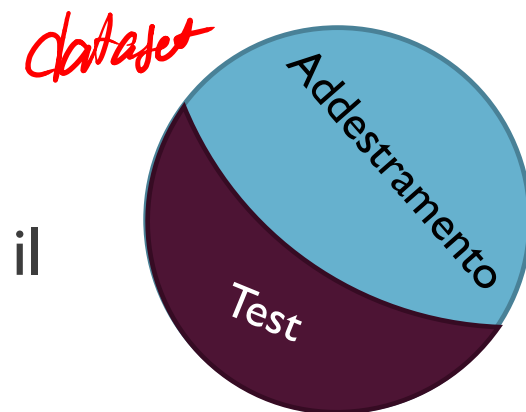
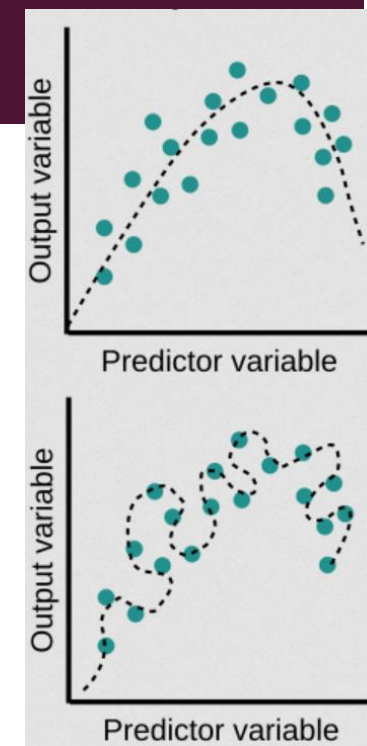
DATASET DI TEST

- Valutare le performance di un modello sui dati di addestramento può essere pericoloso
- Soluzione: suddivido i dati in due porzioni:
 - Dataset di addestramento
 - Dataset di test

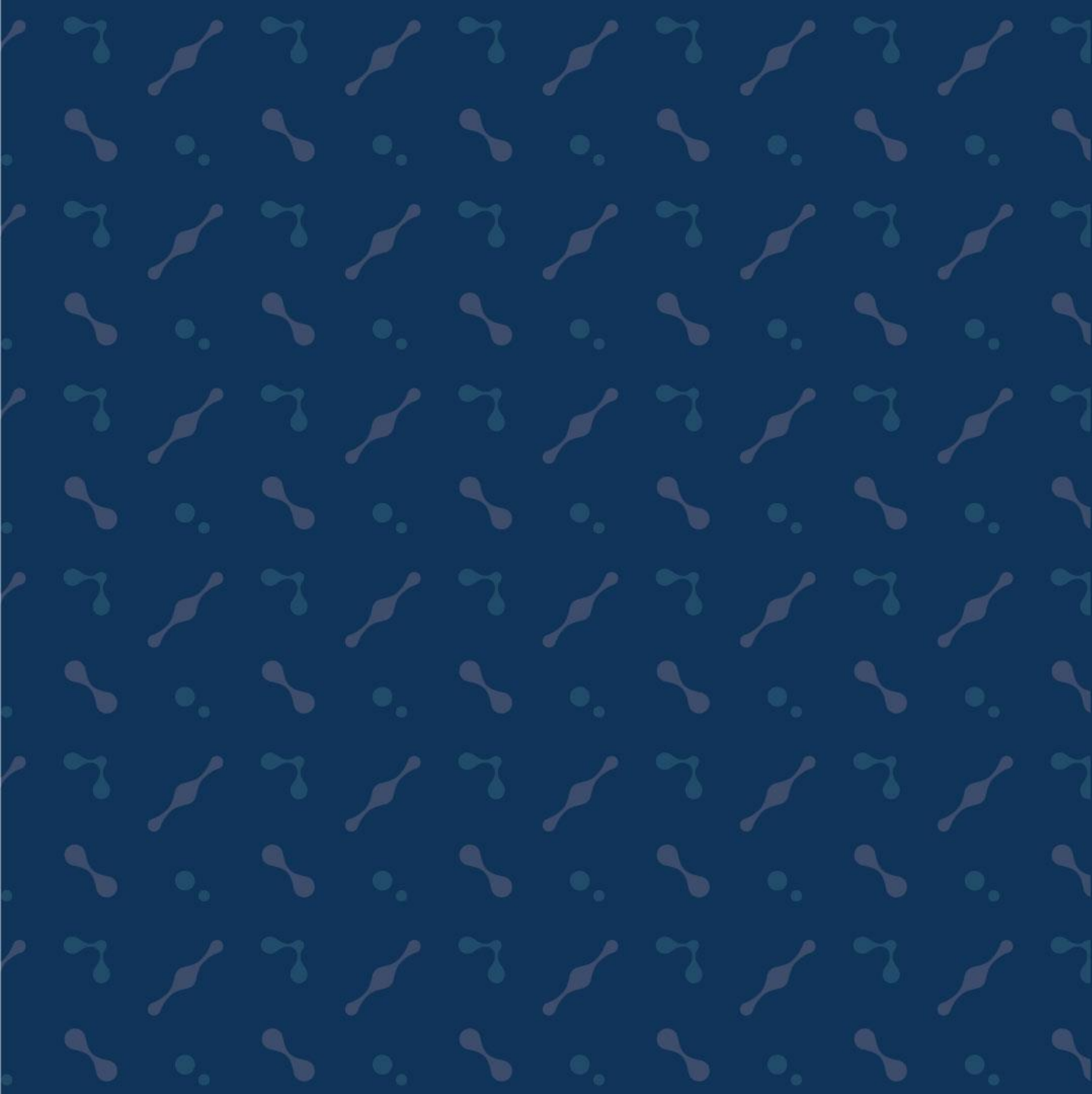
- Addestro il modello sulla base del dataset di addestramento
 - Valuto le performance sulla base del dataset di test per valutare le capacità di **generalizzazione** del modello
- » » » » » voglio un modello che sia in grado di **esprimere buone performance**, senza trascurare la parte di **generalizzazione**

RIASSUMENDO

- La valutazione delle performance di un modello non è un tema banale
- Dobbiamo trovare uno o più **indicatori relativi** che ci permettano di valutare svariate tipologie di modello
 - indipendentemente dal tipo di dati e dal fenomeno che stiamo studiando
- Dobbiamo anche considerare che il modello deve lavorare bene non solo sui dati usati per l'addestramento, ma anche su **potenziali dati mai visti**
- Optiamo per suddividere il dataset in due parti (addestramento/**test**), utilizziamo il dataset di test per valutare il modello (capacità di **generalizzazione**)



$w_1, \dots, w_n, b_1, \dots, b_p$ $w_1, \dots$

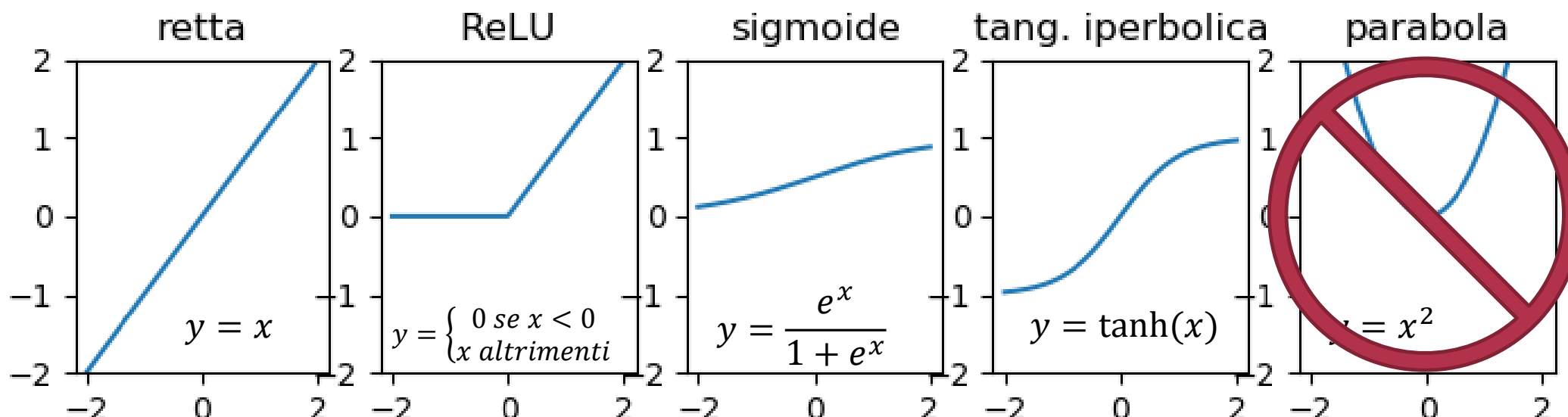


.04

Approfondimenti sulle tecniche delle reti neurali

APPROFONDIMENTO I – FZ. DI ATTIVAZIONE

- La funzione di attivazione è (usualmente) una funzione non-lineare
- La corretta scelta della fz. di attivazione rappresenta **il successo delle reti neurali**

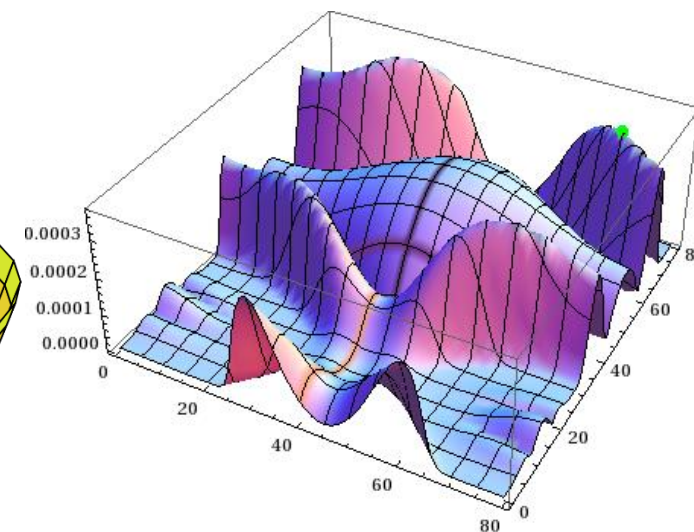
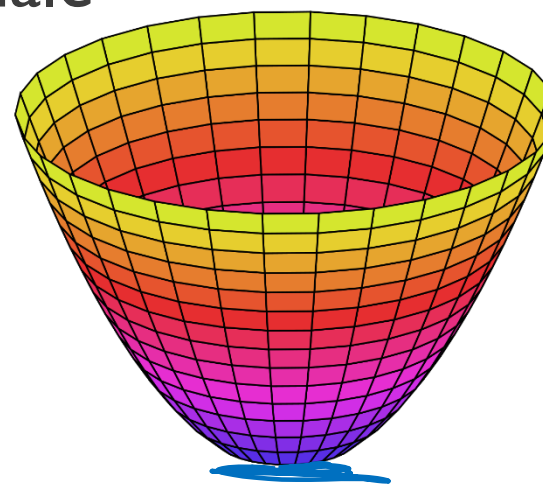


MOMENTO INTERATTIVO (IV) – RETI NEURALI

- Approssimazione del seno
 - https://colab.research.google.com/drive/IBcZOmgPMPUjZ6sIaTIn5WrE-6puVavO_?usp=sharing
- <https://playground.tensorflow.org>
- Notebook rete_neurale_mnist.ipynb da GitHub

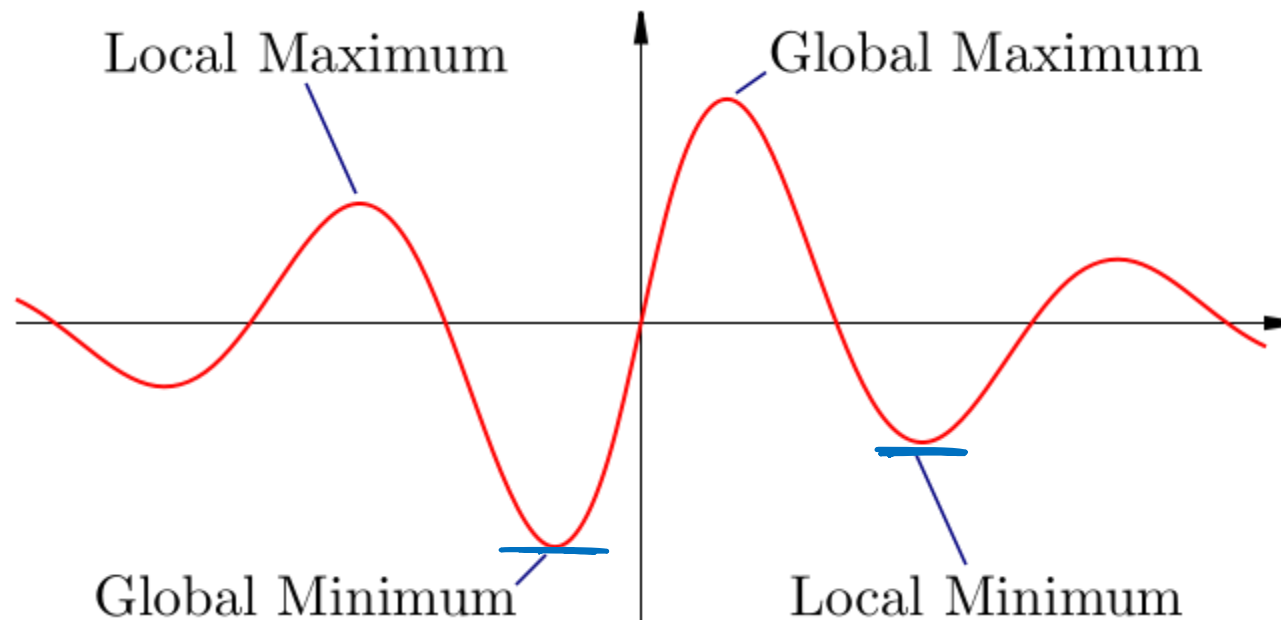
APPROFONDIMENTO II – OTTIMIZZAZIONE (I)

- La rete neurale richiede un tempo notevole di addestramento
- Altre tecniche di machine learning consentono di ottenere modelli in meno di un secondo
- Obiettivo del modello: **minimizzare** (*ottimizzare*) una **funzione di perdita** $\mathcal{L}(y_{reale}, y_{predetto}), w, \dots, w_r, b, \dots, b_r$
- Per alcuni modelli, la soluzione è banale
- Nel caso della rete neurale, non lo è



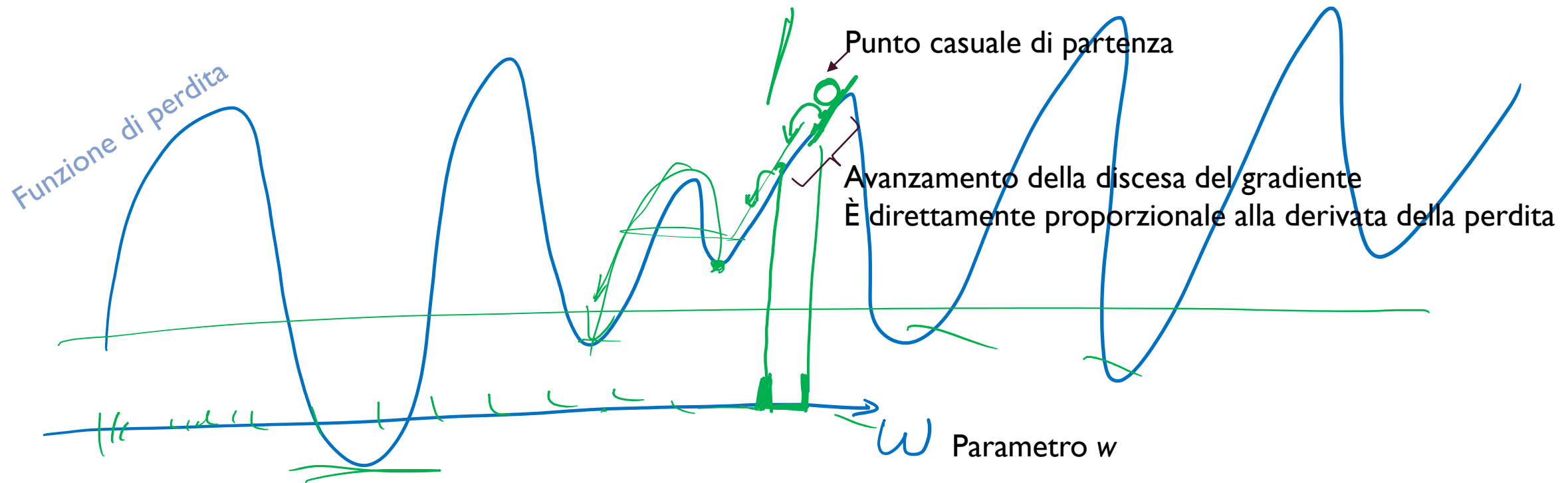
APPROFONDIMENTO II – OTTIMIZZAZIONE (II)

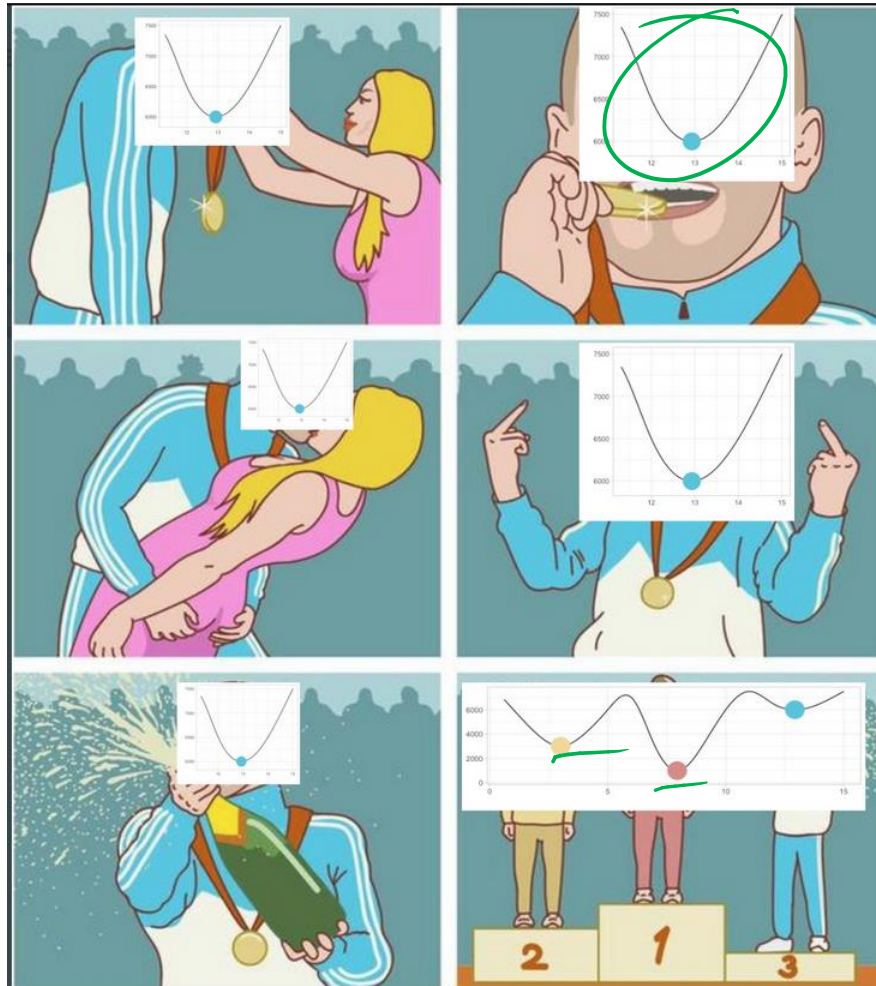
- Per le reti neurali, si utilizza un algoritmo che fornisce...
- Risultati approssimati
- E senza garanzia di produrre il miglior punto di minimo (*globale*)



APPROFONDIMENTO II – SGD – LAVAGNA

- L'algoritmo si chiama DISCESA DEL GRADIENTE





22

376

2.737

Scrivi un nuovo messaggio

berlander

neuroscientist with soft spot for art
PhD, Aalto University
187 following 77 followers
Iscrizione a agosto 2018

David Robinson @drob · 17h
gradient descent



David Robinson
@drob

gradient descent

[Traduci il Tweet](#)

4:46 PM · 11 ago 2021 · Twitter Web App

343 Retweet 33 Tweet di citazione

2.737 Mi piace



Twitta la tua

Rispondi



Naïve Bayesian @naiv... · 16h

In risposta a @drob
Machine learning.



David Robinson
@drob

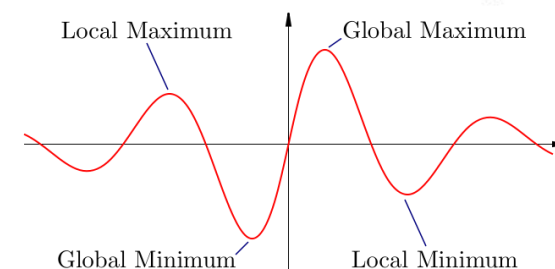
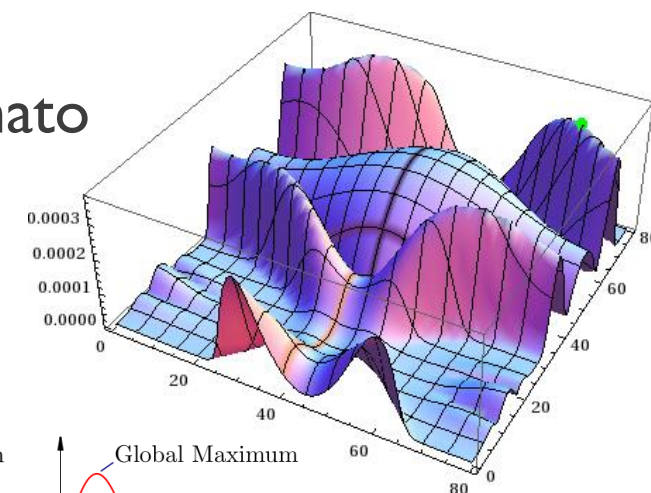
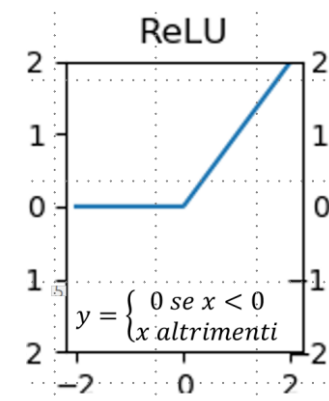
gradient descent

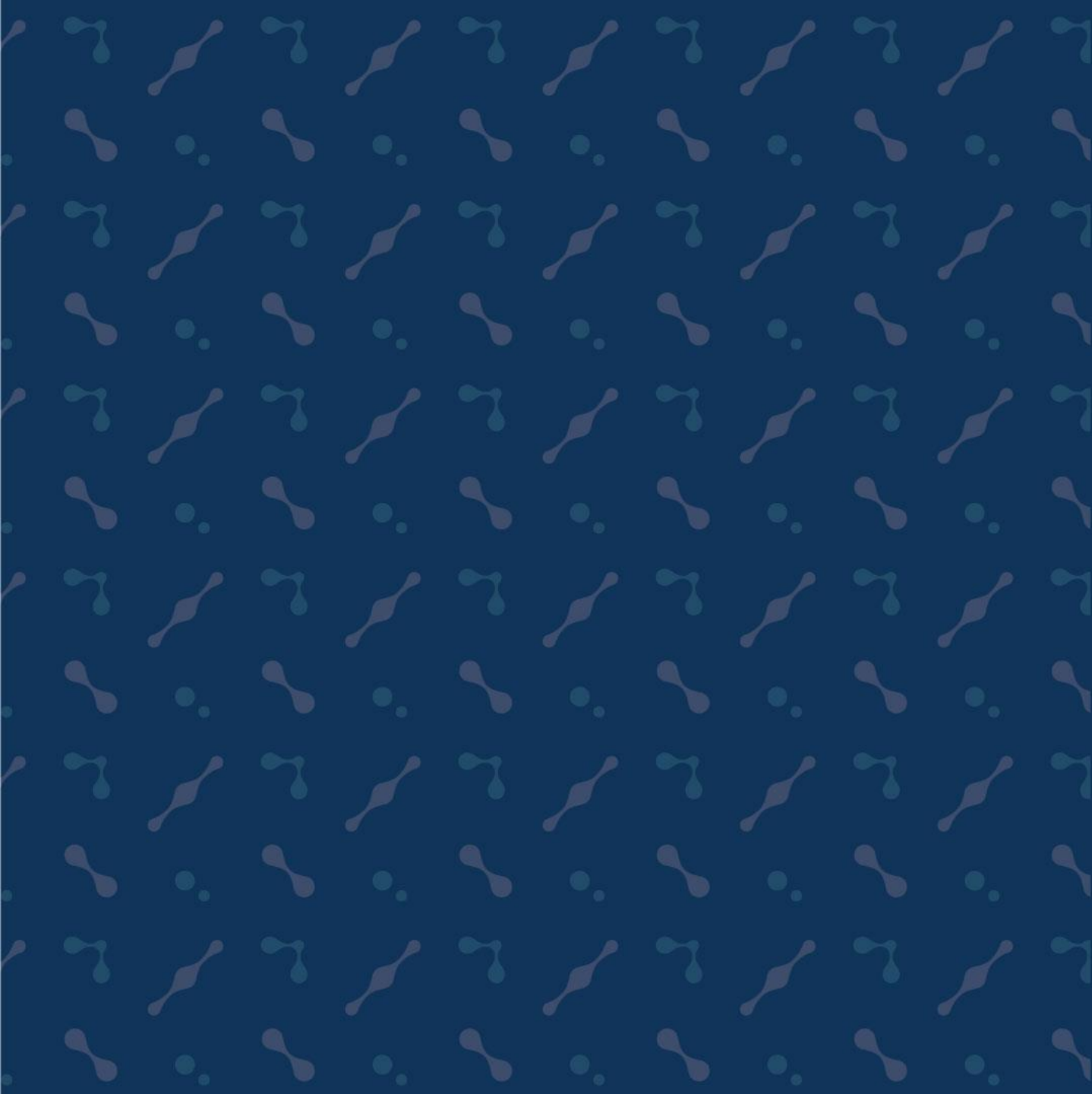
[Translated from English by Google](#)

gradient descent

RIASSUMENDO

- La scelta della funzione di attivazione è un passo importante nella progettazione delle reti neurali
- Vi sono varie scelte possibili; nella visione artificiale si preferisce usare la funzione ReLU (o funzioni simili)
- Il Machine Learning prevede la minimizzazione di una funzione di perdita al fine di ottenere il modello finale
- Per le reti neurali si utilizza un algoritmo molto intuitivo, denominato discesa del gradiente
- Partendo da una configurazione casuale dei pesi della rete, raffigurabile come un punto di uno spazio a molte dimensioni...
- ...si discende questo spazio a piccoli passi...
- ...seguendo ogni volta la direzione di massima pendenza
- C'è il rischio di rimanere «bloccati» in minimi locali (→ configurazione dei pesi non ottimale)





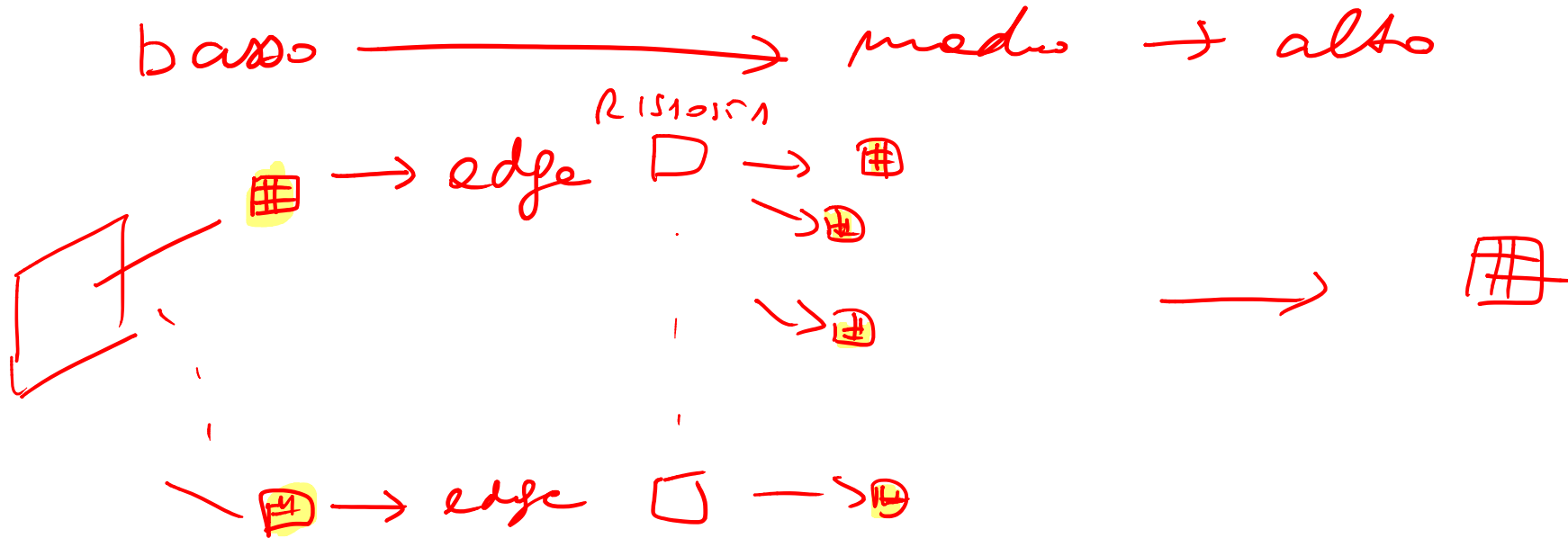
.05

**Applicazioni di reti
neurali alle immagini**

OK, MA, I FILTRI?

- Le reti neurali classiche sono troppo inefficienti per poter essere applicate a immagini (al di fuori di pochi esempi didattici)
- C'è un'architettura che utilizza le cross-correlazioni ed è molto più indicata per lavorare sulle immagini
- RETI NEURALI CONVOLUZIONALI

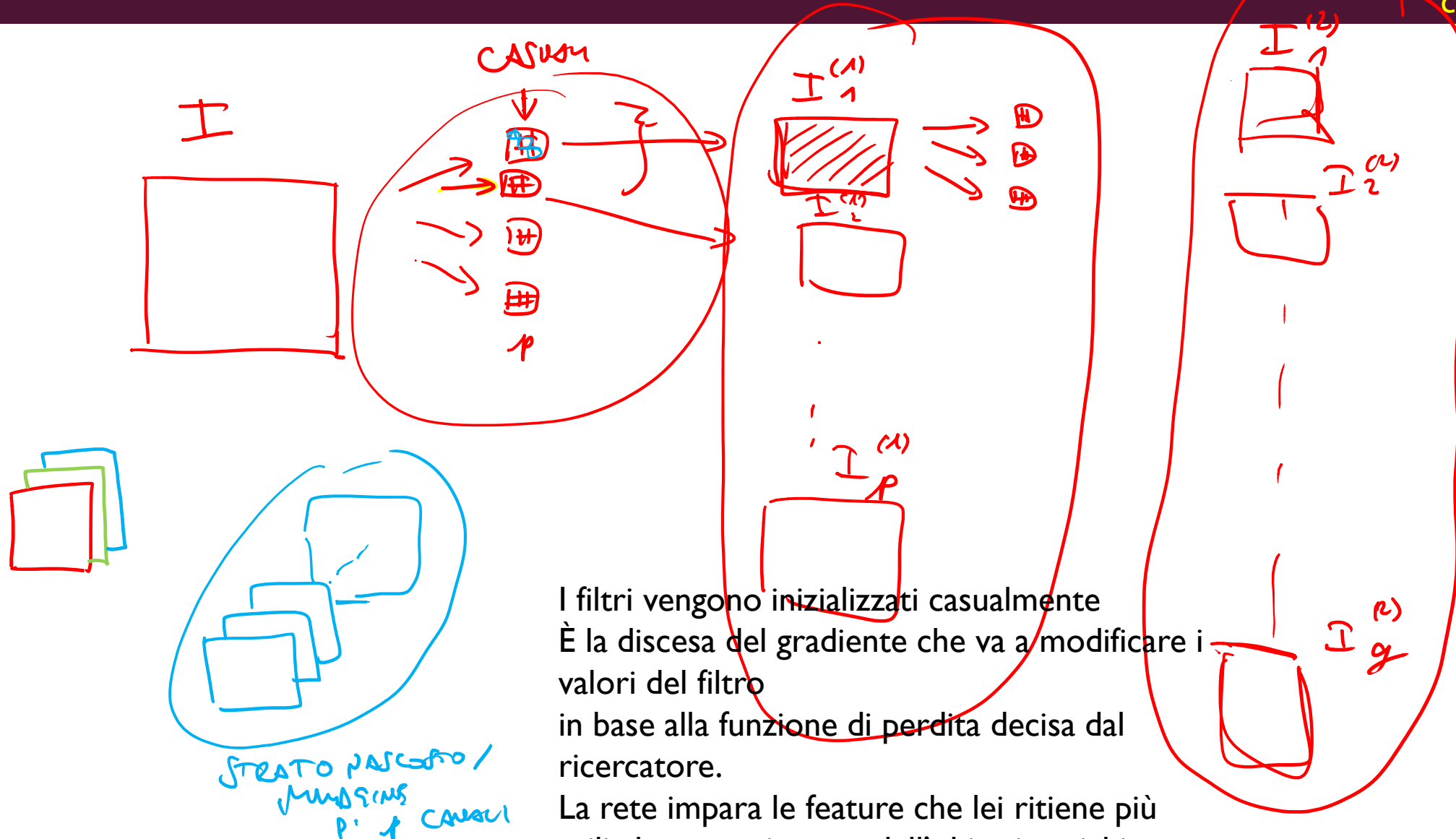
WORKFLOW CLASSICO



Tutti i filtri di correlazione/convoluzione sono decisi dallo scienziato ai fini di ottenere una risposta a determinati set di feature

LA RETE CONVOLUZIONALE

Strato nascosto
convoluzionale



RIASSUMENDO

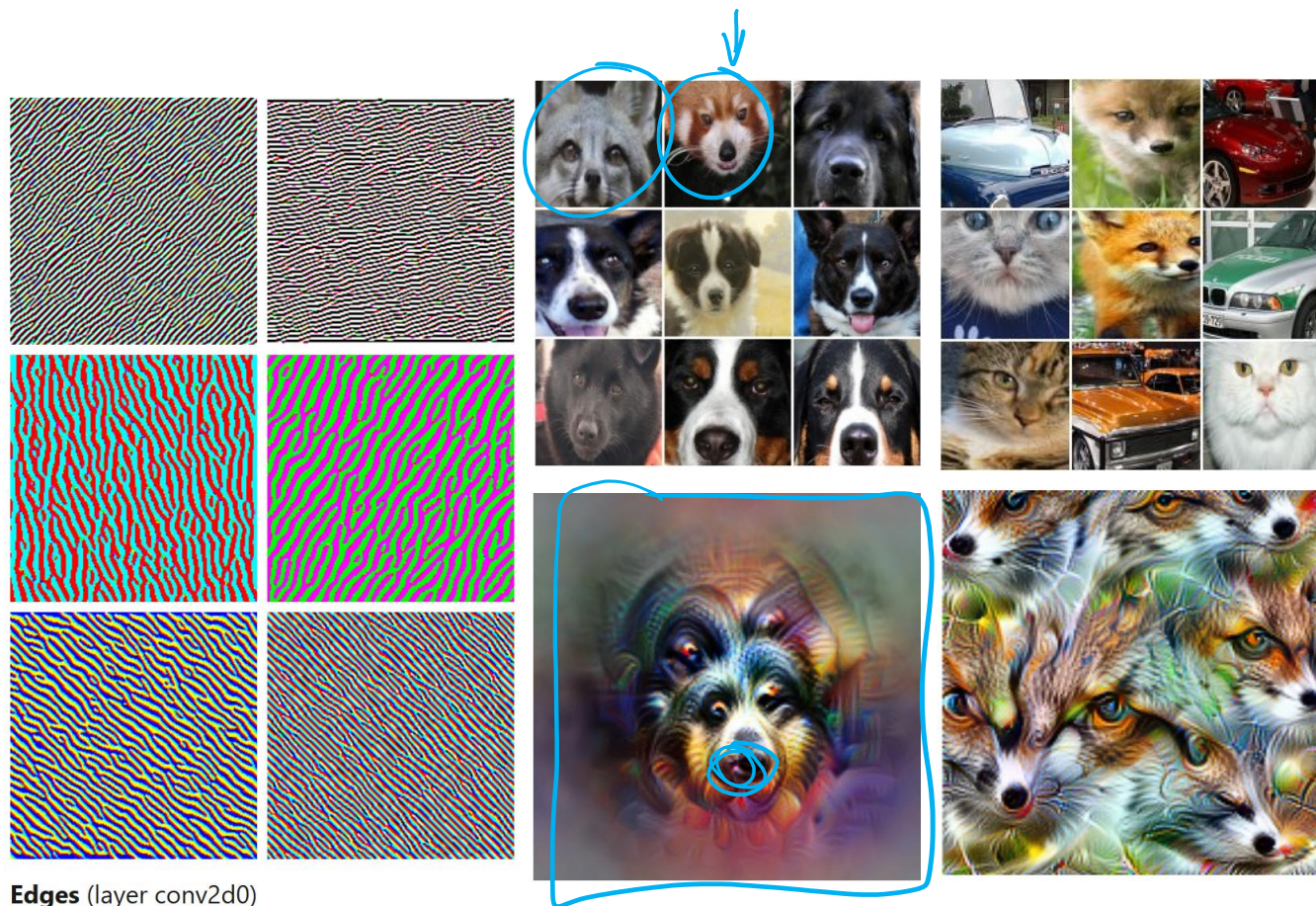
- Le reti neurali convoluzionali si basano sull'idea che le *features* sono ordinabili in una gerarchia, dal basso livello all'alto livello
 - NB: basso livello = bordi, colori; alto livello = figure complesse
- Workflow «classico»: io esperto progetto le convoluzioni in modo che i filtri identifichino le *feature* che io voglio trovare
- Workflow «moderno»: iniziamo da filtri casuali, lasciamo che sia il programma ad imparare quali siano le *feature* da ricercare
 - Si tratta di una rete neurale in cui c'è una connettività ridotta che aiuta l'addestramento e la velocità dei calcoli

FEATURE NELLE CNN

- Analizzando i filtri prodotti dalle CNN, si è visto...
- Che i filtri dei primi strati identificano feature di basso livello
- I filtri degli strati più alti uniscono le feature di basso livello in feature di alto livello

Vedi:

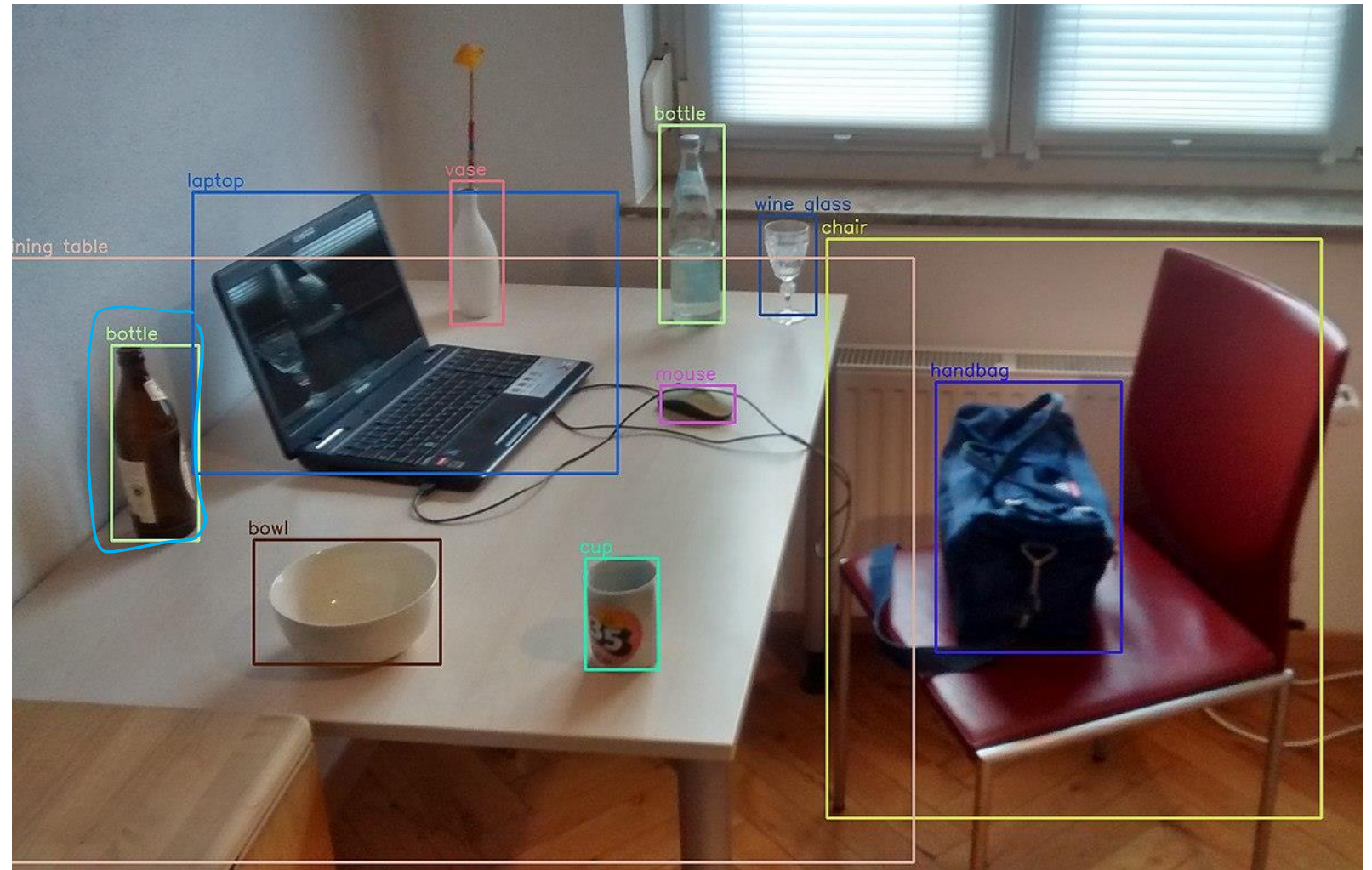
<https://distill.pub/2017/feature-visualization/>



Edges (layer conv2d0)

INDIVIDUAZIONE DI OGGETTI

- L'individuazione di oggetti è una classificazione locale dell'immagine
- Addestro una rete a riconoscere n oggetti
- Idea di massima: applico la rete a delle *patch* dell'immagine per vedere se esiste un oggetto all'interno
- Classificazione (WHAT) + Localizzazione (WHERE)



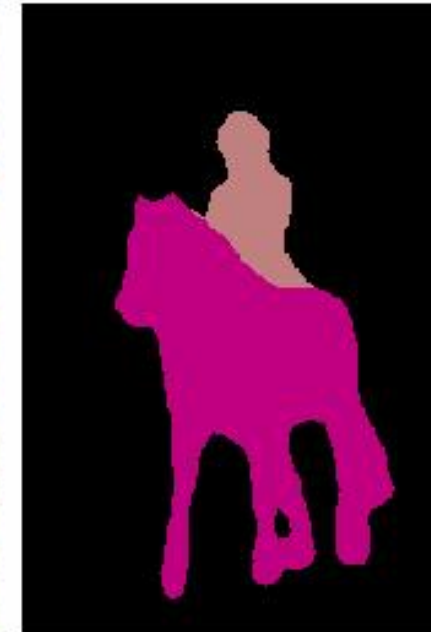
MOMENTO INTERATTIVO (V) – YOLO

- <https://github.com/marcozullich/object-detection-opencv>
- Riconoscimento mascherine:
<https://github.com/benedettaliberatori/Modified-Yolov4Tiny-RaspberryPi/>

SEGMENTAZIONE D'IMMAGINI

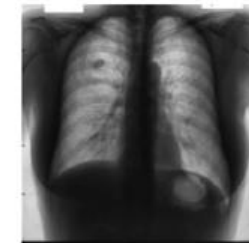
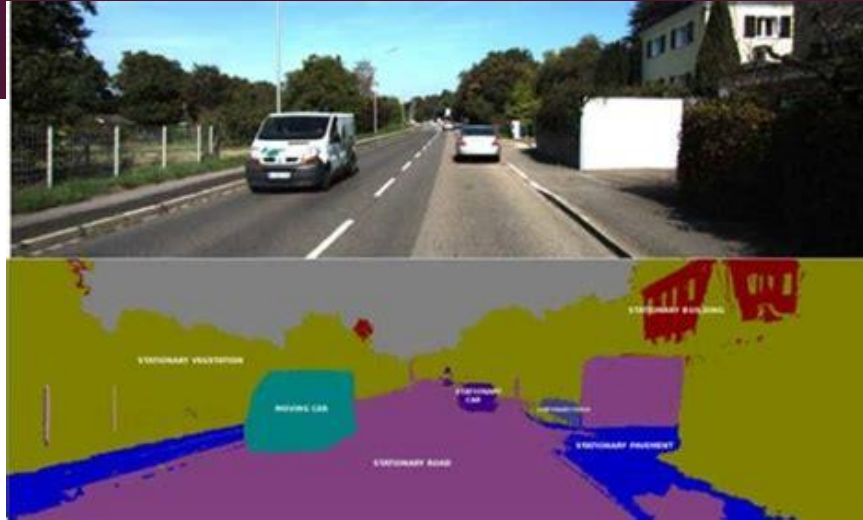
- Il riconoscimento di oggetti fornisce informazioni approssimative sulla localizzazione di questi ultimi
- La segmentazione si occupa di ritagliare i bordi (*determinare i segmenti*) dove gli oggetti risiedono

Idea di base: anziché classificare l'intera immagine, **voglio classificare ogni singolo pixel**

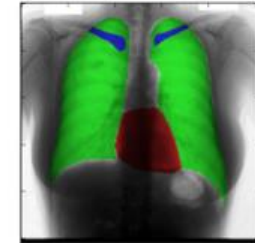


APPLICAZIONI

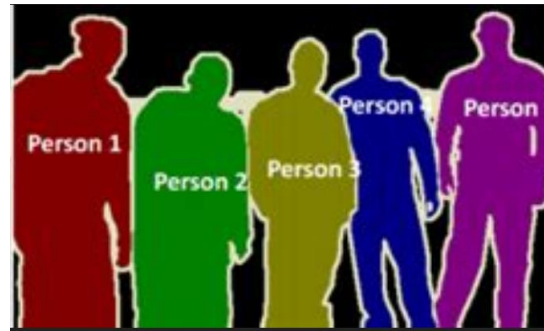
- Guida autonoma
- Medicina
- Videosorveglianza
- ...



Input Image



Segmented Image

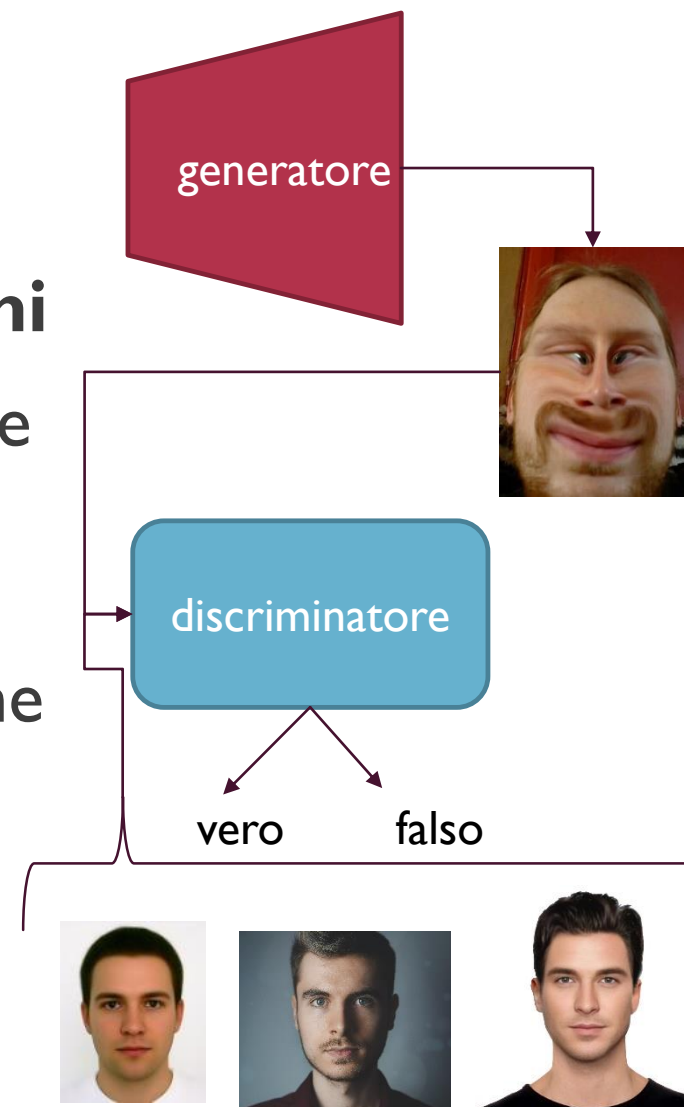


MOMENTO INTERATTIVO (VI) - SEGMENTAZIONE

- Notebook segmentation.ipynb da repo GitHub

GAN

- Le Reti Generative Avversarie sono modelli composti da 2 reti neurali
- Una rete (*generatore*) si occupa di **generare immagini**
- La seconda (*discriminatore*) **valuta** le immagini generate dalla prima, cercando di determinare se queste sono **vere o false (sintetiche)**
- Il discriminatore è addestrato sia con immagini reali che con immagini sintetiche
- Lo scopo del generatore è quello di produrre immagini che *sbugiardino* il discriminatore



GENERAZIONE D'IMMAGINI

- L'applicazione più semplice è la creazione di immagini senza *suggerimento*, ovvero, non diamo alcun input al modello, lo lasciamo che generi un'immagine
- Generazione di volti realistici:
<https://thispersondoesnotexist.com>

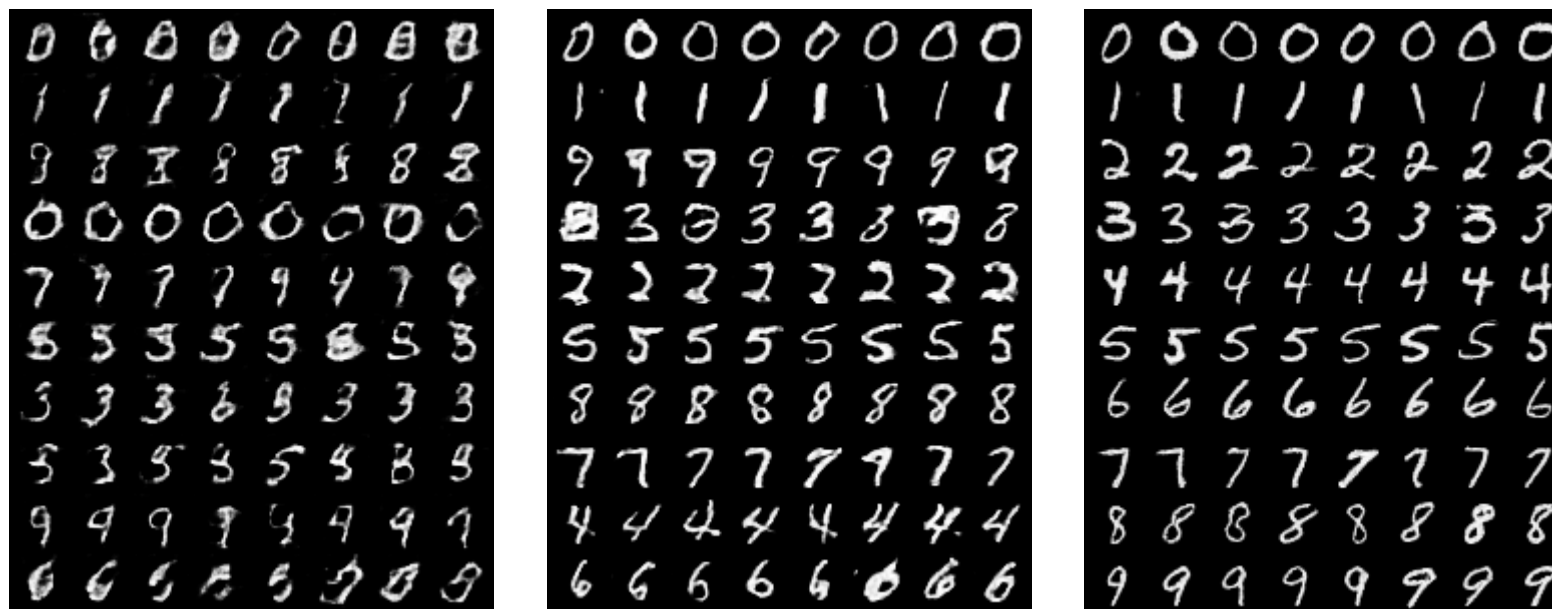


GENERAZIONE CONDIZIONATA

- La generazione può anche essere condizionata, esempio ad una determinata categoria di oggetti (GAN CONDIZIONALE)

- Generazione di cifre scritte a mano:

<https://github.com/marcozullich/Pytorch-conditional-GANs>



APPLICAZIONI

- Arte
- Generazione di loghi
- Generazione di immagini non coperte da copyright
- Rispetto privacy →
- *Data Augmentation*
- ...



GAN 4 GAMING

- Le GAN possono essere utilizzate nel gaming per generare oggetti da parte degli sviluppatori
 - Generazione livelli Super Mario: <https://youtu.be/CbifnDFguD8?t=56>
 - «Svecchiamento» vecchie texture: <https://towardsdatascience.com/generative-adversarial-networks-revitalizing-old-video-game-textures-669493f883a0>
 - Abbellimento paesaggi Minecraft:
<https://twitter.com/matthen2/status/1564192185909739521?t=7ypdhLolaF9yijH5Gj88Og&s=35>

STYLE TRANSFER

- Il trasferimento di stile è una generazione guidata di immagini che consente di applicare un determinato stile artistico ad un'immagine di partenza
- Possiamo vederlo come una GAN condizionale in cui la condizione è data dall'immagine di partenza (ed eventualmente dallo stile scelto)
- ArcaneGAN:
<https://huggingface.co/spaces/akhaliq/ArcaneGAN>



IMAGE-TO-TEXT

- Image-to-text = generare del testo da un'immagine
- Che tipo di testo?
- Idea più comune: image captioning, ovvero generare una didascalia/descrizione del contenuto di un'immagine
- Demo: <https://milhidaka.github.io/chainer-image-caption/>



View of mediterranean / spanish home with a dock and water view

APPLICAZIONI

- Sussidio a non vedenti e ipovedenti
 - Creazione automatica didascalie
 - Creazione automatica sottotitoli (video-to-text)
- Assistente vocale guida veicoli
- ...

TEXT-TO-IMAGE

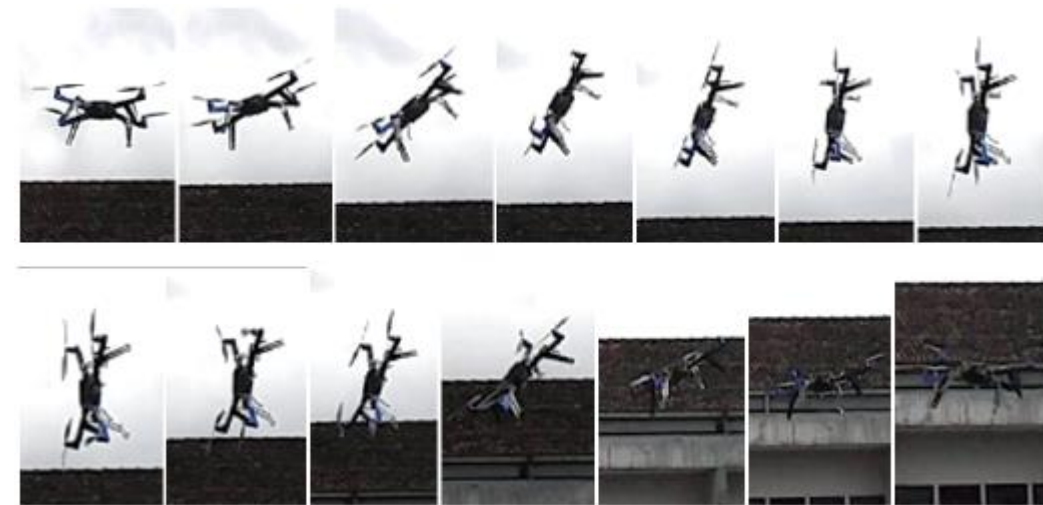
- Esempio più interessante: vogliamo che l'AI generi un'immagine sulla base del testo che noi le forniamo
- Il campo è diventato molto mainstream negli ultimi 2 anni grazie a DALL•E e Imagen
- Demo: <https://www.craiyon.com/>
- Demo (con accesso ristretto): <https://labs.openai.com>

«Jack Sparrow in "Arancia Meccanica"»



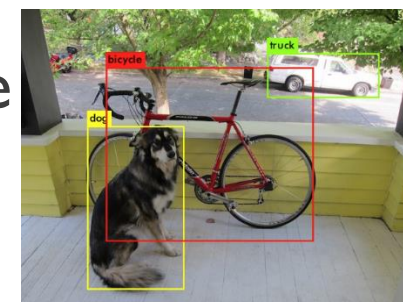
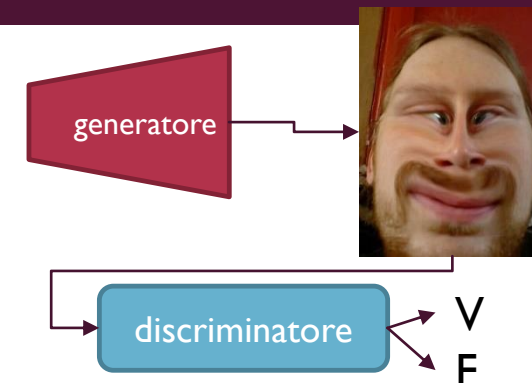
TEXT-TO-VIDEO

- Possiamo anche applicare la medesima tecnologia alla generazione di video
- Il video è una sequenza di immagini (più una traccia audio opzionale)
- Demo molto interessante: generazione di volti parlanti con possibilità di traduzione:
<https://huggingface.co/spaces/CVPR/ml-talking-face>



RIASSUMENDO

- Le GAN sono un approccio generativo alle reti neurali
 - Ho due reti anziché una: un **generatore** (che genera le immagini) e un **discriminatore** (che distingue fra immagine reale/sintetica)
 - Obiettivo: il generatore deve **imbrogliare** il discriminatore
- La **individuazione di oggetti** si occupa di riconoscere istanze di oggetti noti e di **localizzarli** nell'immagine
- La segmentazione d'immagine si occupa di localizzare accuratamente nell'immagine oggetti noti
- Esistono tecniche image-to-text e text-to-image che permettono di generare del testo da un'immagine e immagine da testo rispettivamente



Grazie dell'attenzione!

ai.units.it

github.com/marcozullich

www.zullich.it

marco.zullich@phd.units.it



ARTIFICIAL INTELLIGENCE
& DATA ANALYTICS