




DALLA TEORIA AL CODICE IN PYTHON

# COMPUTER VISION & DEEP LEARNING

UN LIBRO PER INIZIARE A ENTRARE IN  
QUESTI MONDI

# Computer Vision e Deep Learning - Cosa è la Visione Artificiale e cosa centra il Deep Learning

---

 [intelligenzaartificialeitalia.net/post/computer-vision-e-deep-learning-cosa-è-la-visione-artificiale-e-cosa-centra-il-deep-learning](https://intelligenzaartificialeitalia.net/post/computer-vision-e-deep-learning-cosa-è-la-visione-artificiale-e-cosa-centra-il-deep-learning)

Vorrei iniziare questo articolo andando a vedere la definizione di computer vision

## Cosa è la Computer Vision :

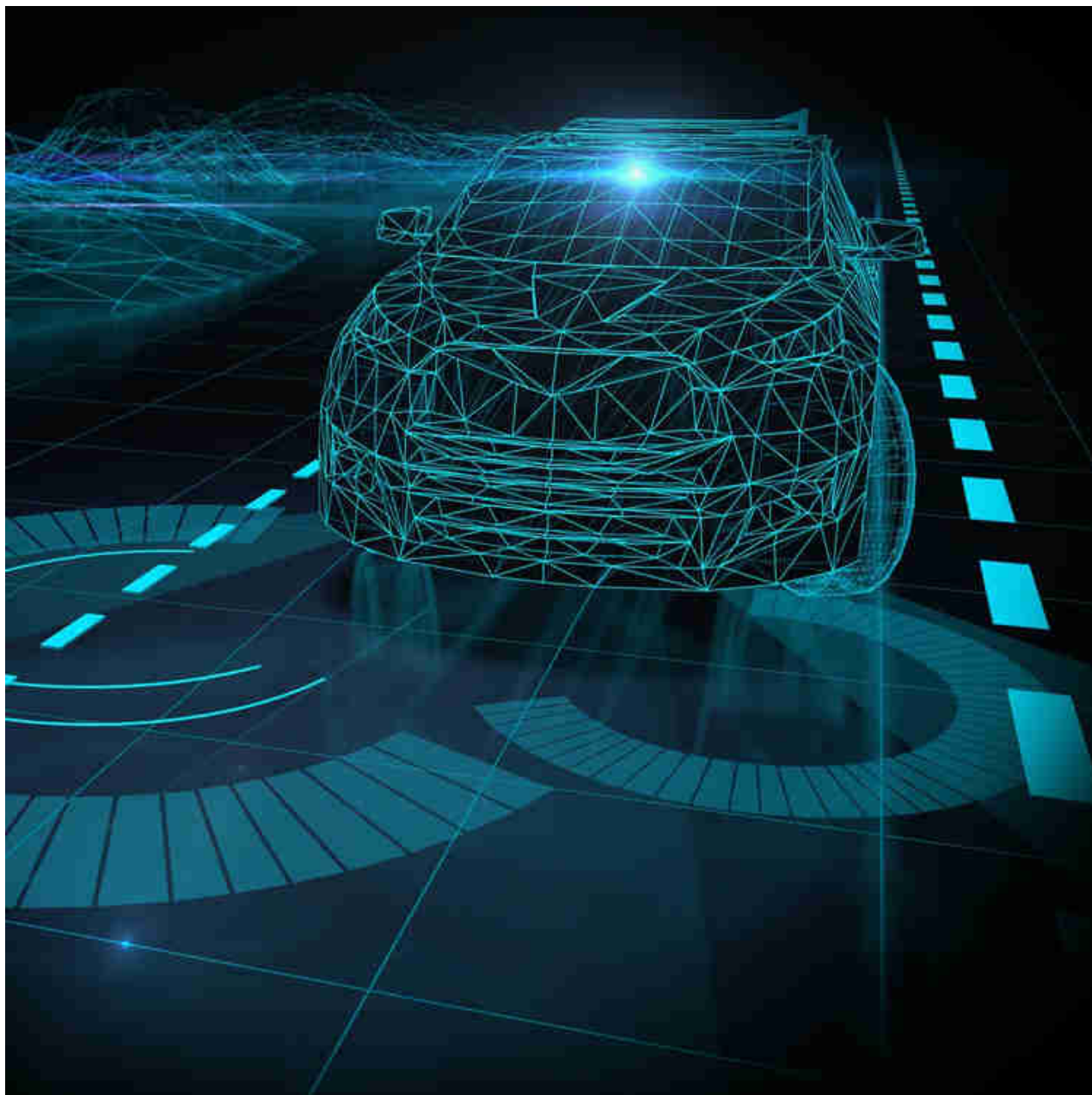
---

La computer vision (CV) è il campo scientifico che definisce il modo in cui le macchine interpretano il significato di immagini e video. Gli algoritmi di visione artificiale analizzano determinati criteri in immagini e video, quindi applicano interpretazioni a compiti predittivi o decisionali.

Oggi, le tecniche di deep learning sono più comunemente utilizzate per la visione artificiale. Questo articolo esplora diversi modi in cui puoi utilizzare il deep learning per la visione artificiale. In particolare, imparerai i vantaggi dell'utilizzo delle reti neurali convoluzionali (CNN), che forniscono un'architettura multistrato che consente alle reti neurali di concentrarsi sulle caratteristiche più rilevanti nell'immagine.

### In questo articolo imparerai:

- [Cos'è la visione artificiale?](#)
- [Architetture di deep learning per CV](#)
- [Usi del deep learning nella computer vision](#)



Apprendimento profondo per la visione artificiale - Deep Learning per la Computer Vision

## **Che cos'è la visione artificiale o Computer Vision?**

---

La computer vision è una sotto-area di machine learning dedicata all'interpretazione e alla comprensione di immagini e video. Viene utilizzato per insegnare ai computer a "vedere" e a utilizzare le informazioni visive per eseguire attività visive che gli esseri umani possono.

I modelli di visione artificiale sono progettati per tradurre i dati visivi in base alle caratteristiche e alle informazioni contestuali identificate durante l'addestramento. Ciò consente ai modelli di interpretare immagini e video e applicare tali interpretazioni a compiti predittivi o decisionali.

Sebbene entrambi siano correlati ai dati visivi, l'elaborazione delle immagini non è la stessa della visione artificiale. L'elaborazione delle immagini comporta la modifica o il miglioramento delle immagini per produrre un nuovo risultato. Può includere l'ottimizzazione della luminosità o del contrasto, l'aumento della risoluzione, la sfocatura delle informazioni sensibili o il ritaglio. La differenza tra elaborazione delle immagini e visione artificiale è che la prima non richiede necessariamente l'identificazione del contenuto.

**Reti neurali convoluzionali:** le fondamenta della moderna visione artificiale I moderni algoritmi di visione artificiale si basano su reti neurali convoluzionali (CNN), che forniscono un notevole miglioramento delle prestazioni rispetto ai tradizionali algoritmi di elaborazione delle immagini.

Le CNN sono reti neurali con un'architettura multilivello che viene utilizzata per ridurre gradualmente dati e calcoli al set più pertinente. Questo set viene quindi confrontato con dati noti per identificare o classificare l'input dei dati.

Le CNN vengono generalmente utilizzate per attività di visione artificiale, sebbene possano essere eseguite anche analisi del testo e analisi dell'audio. Una delle prime architetture della CNN è stata AlexNet (descritta di seguito), che ha vinto la sfida di riconoscimento visivo di ImageNet nel 2012.

**Come funzionano le CNN** Quando un'immagine viene elaborata da una CNN, ogni colore di base utilizzato nell'immagine (ad es. rosso, verde, blu) viene rappresentato come una matrice di valori. Questi valori vengono valutati e condensati in tensori 3D (nel caso di immagini a colori), che sono raccolte di pile di mappe di caratteristiche legate a una sezione dell'immagine. Questi tensori vengono creati facendo passare l'immagine attraverso una serie di strati convoluzionali e di raggruppamento, che vengono utilizzati per estrarre i dati più rilevanti da un segmento di immagine e condensarli in una matrice più piccola e rappresentativa. Questo processo viene ripetuto numerose volte (a seconda del numero di strati convoluzionali nell'architettura). Le caratteristiche finali estratte dal processo convoluzionale vengono inviate a un livello completamente connesso, che genera previsioni.

## Architetture di deep learning per la visione artificiale

---

Le prestazioni e l'efficienza di una CNN sono determinate dalla sua architettura. Ciò include la struttura dei livelli, come sono progettati gli elementi e quali elementi sono presenti in ogni livello. Sono state create molte CNN, ma i seguenti sono alcuni dei progetti più efficaci.

### AlexNet (2012)

---

AlexNet è un'architettura basata sulla precedente architettura LeNet. Comprende cinque strati convoluzionali e tre strati completamente connessi. AlexNet utilizza una struttura a doppia pipeline per consentire l'uso di due GPU durante l'addestramento. La principale differenza tra AlexNet e le architetture precedenti è l'uso di unità lineari rettificate (ReLU) invece delle funzioni di attivazione sigmoide o Tanh utilizzate nelle reti neurali tradizionali. ReLU è più semplice e veloce da calcolare, consentendo ad AlexNet di addestrare i modelli più velocemente.

## **GoogleNet (2014)**

---

GoogleNet, noto anche come Inception V1, si basa sull'architettura LeNet. È composto da 22 strati formati da piccoli gruppi di circonvoluzioni, detti "moduli di inizio". Questi moduli di avvio utilizzano la normalizzazione batch e RMSprop per ridurre il numero di parametri che GoogleNet deve elaborare. RMSprop è un algoritmo che utilizza metodi di tasso di apprendimento adattivo.

## **VGGNet (2014)**

---

VGG 16 è un'architettura a 16 livelli (alcune varianti avevano 19 livelli). VGGNet ha livelli convoluzionali, un livello di pool, alcuni livelli convoluzionali in più, un livello di pool, molti altri livelli di conv e così via. VGG si basa sulla nozione di una rete molto più profonda con filtri più piccoli: utilizza circonvoluzioni  $3 \times 3$  fino in fondo, che è la dimensione del filtro conv più piccola che esamina solo alcuni dei pixel vicini. Utilizza piccoli filtri a causa di un minor numero di parametri, rendendo possibile l'aggiunta di più livelli. Ha lo stesso campo ricettivo effettivo di uno strato convoluzionale  $7 \times 7$ .

## **ResNet (2015)**

---

ResNet, acronimo di Residual Neural Network, è un'architettura progettata per avere un gran numero di livelli: le architetture tipicamente utilizzate vanno da ResNet-18 (con 18 livelli) a ResNet-1202 (con 1202 livelli). Questi livelli sono impostati con gated unità o "salta connessioni" che gli consentono di passare informazioni a livelli convoluzionali successivi. ResNet utilizza anche la normalizzazione batch per migliorare la stabilità della rete.

## **Xception (2016)**

---

Xception è un'architettura basata su Inception, che sostituisce i moduli di inizio con circonvoluzioni separabili in profondità (circonvoluzione in profondità seguita da circonvoluzioni puntuali). Funziona catturando prima le correlazioni tra mappe di

funzionalità e poi le correlazioni spaziali. Ciò consente un uso più efficiente dei parametri del modello.

## ResNeXt-50 (2017)

---

ResNeXt-50 è un'architettura basata su moduli con 32 percorsi paralleli. Utilizza la cardinalità per ridurre gli errori di convalida e rappresenta una semplificazione dei moduli di inizio utilizzati in altre architetture.

## Usi del deep learning nella visione artificiale

---

Lo sviluppo delle tecnologie di deep learning ha consentito la creazione di modelli di computer vision più accurati e complessi. Con l'aumento di queste tecnologie, l'incorporazione di applicazioni di visione artificiale sta diventando più utile. Di seguito sono riportati alcuni modi in cui il deep learning viene utilizzato per migliorare la visione artificiale.







Apprendimento profondo per la visione artificiale - Deep Learning per la Computer Vision

## Rilevamento di oggetti e Visione Artificiale

---

Esistono due tipi comuni di rilevamento di oggetti eseguiti tramite tecniche di visione artificiale:

- **Rilevamento di oggetti in due fasi** : il primo passaggio richiede una rete di proposte regionali (RPN), che fornisce un numero di regioni candidate che possono contenere oggetti importanti. Il secondo passaggio consiste nel passare le proposte della regione a un'architettura di classificazione neurale, comunemente un algoritmo di raggruppamento gerarchico basato su RCNN o un pool di regioni di interesse (ROI) in Fast RCNN. Questi approcci sono abbastanza precisi, ma possono rallentare molto.
- **Rilevamento di oggetti in un solo passaggio** : con la necessità di rilevamento di oggetti in tempo reale, sono emerse architetture di rilevamento di oggetti in un solo passaggio, come YOLO, SSD e RetinaNet. Questi combinano la fase di rilevamento e classificazione, regredendo le previsioni del riquadro di delimitazione. Ogni riquadro di delimitazione è rappresentato con poche coordinate, facilitando la combinazione della fase di rilevamento e classificazione e velocizzando l'elaborazione.

---

## Localizzazione e rilevamento di oggetti

La localizzazione dell'immagine viene utilizzata per determinare dove si trovano gli oggetti in un'immagine. Una volta identificati, gli oggetti sono contrassegnati da un riquadro di delimitazione. Il rilevamento degli oggetti si estende su questo e classifica gli

oggetti identificati. Questo processo si basa su CNN come AlexNet, Fast RCNN e Faster RCNN.

La localizzazione e il rilevamento di oggetti possono essere utilizzati per identificare più oggetti in scene complesse. Questo può quindi essere applicato a funzionalità come l'interpretazione di immagini diagnostiche in medicina.

## **Segmentazione semantica**

---

La segmentazione semantica, nota anche come segmentazione degli oggetti, è simile al rilevamento degli oggetti tranne per il fatto che si basa sui pixel specifici relativi a un oggetto. Ciò consente di definire più accuratamente gli oggetti immagine e non richiede riquadri di delimitazione. La segmentazione semantica viene spesso eseguita utilizzando reti completamente convoluzionali (FCN) o U-Net.

Un uso popolare per la segmentazione semantica è per l'addestramento di veicoli autonomi. Con questo metodo, i ricercatori possono utilizzare immagini di strade o passaggi con confini accuratamente definiti per gli oggetti.

## **Stima della posa**

---

La stima della posa è un metodo utilizzato per determinare dove si trovano le articolazioni in un'immagine di una persona o di un oggetto e cosa indica il posizionamento di tali articolazioni. Può essere utilizzato sia con immagini 2D che 3D. L'architettura principale utilizzata per la stima della posa è PoseNet, che si basa sulle CNN.

La stima della posa viene utilizzata per determinare dove parti del corpo possono apparire in un'immagine e può essere utilizzata per generare posizioni realistiche o movimenti di figure umane. Spesso, questa funzionalità viene utilizzata per la realtà aumentata, il rispecchiamento dei movimenti con la robotica o l'analisi del cammino.

**Sei Interessato a sapere qualcosa nello specifico sulla Visione Artificiale ?**

**Scrivi un commento qui sotto !**



# Semplici progetti di Deep Learning e Computer Vision Con Esempi pratici in Python e OpenCV

---

 [intelligenzaartificialeitalia.net/post/semplci-progetti-di-deep-learning-e-computer-vision-con-esempi-pratici-in-python-e-opencv](https://intelligenzaartificialeitalia.net/post/semplci-progetti-di-deep-learning-e-computer-vision-con-esempi-pratici-in-python-e-opencv)

## Non hai mai sentito prima la Libreria OpenCv ?

### Ti spieghiamo subito a cosa serve e che progetti permette di fare

OpenCV è un ottimo strumento per l'elaborazione delle immagini e l'esecuzione di attività di visione artificiale. È una libreria open source che può essere utilizzata per eseguire attività come il rilevamento dei volti, il rilevamento delle obiezioni, il rilevamento dei punti di riferimento e molto altro. Supporta più lingue tra cui Python, Java C++. Anche se, per questo articolo, ci limiteremo solo a Python.

La libreria è dotata di centinaia di funzioni e algoritmi utili, tutti a nostra disposizione gratuitamente. Alcune di queste funzioni sono molto comuni e vengono utilizzate in quasi tutte le attività di visione artificiale. Mentre molte delle funzioni sono ancora inesplorate e non hanno ancora ricevuto molta attenzione.

In questo articolo, vedremo alcune delle funzioni più sorprendenti di OpenCV che puoi utilizzare per molte fantastiche applicazioni.

## Prerequisiti :

---

- Se non sai perchè utilizzeremo python, [clicca qui](#)
- Se non hai ancora installato Python, [clicca qui](#)
- Se non sai come scaricare e gestire le librerie, [clicca qui](#)
- Se non sai cosa sia un Dataset, [clicca qui](#)

## Indice :

---

1. [Rimozione Watermark dalle immagini](#)
2. [Rimozione lo sfondo dalle immagini](#)

3. Creazione di filtri per le immagini ( Tipo Instagram o Facebook).

4. Trasformare in cartone qualsiasi immagine



Semplici progetti di Deep Learning e Computer Vision Con Esempi pratici in Python e OpenCV

**Rimuovere Watermark usando OpenCV, Python**

---

Le filigrane potrebbero essere davvero fastidiose e arriva un momento nella vita di tutti in cui cerchiamo in tutto Internet un programma per rimuovere quel piccolo pezzo di testo. Per fortuna, la tua ricerca finisce qui. Usando OpenCV, possiamo consentire questo compito con meno di 10 righe di codice.

Iniziamo importando le librerie richieste (che sono OpenCV e Numpy) e leggendo l'immagine, contenente la filigrana. Quindi cambiamo lo spazio colore da BGR a HSV. Stiamo utilizzando il formato colore HSV in quanto aiuta a ottenere un'immagine maschera migliore che creeremo nel passaggio successivo. Una maschera è un'immagine che ha solo due valori che sono bianco e nero. Lo scopo della maschera è applicare modifiche solo nelle parti specifiche di un'immagine. Per il nostro scopo, l'immagine della maschera dovrebbe contenere valori bianchi ovunque sia presente la filigrana e neri ovunque.

Per creare la nostra immagine maschera, utilizzeremo la sogliatura che ho già eseguito per questa immagine. Che cosa fa la soglia è, dice se un valore si trova nell'intervallo dato o meno. Se desideri utilizzare un'altra immagine, puoi trovare i tuoi valori di soglia utilizzando [questo](#) pezzo di codice. Ora, abbiamo l'immagine della maschera, possiamo chiamare la funzione principale richiesta per eseguire questo compito. Useremo la funzione `cv2.inpaint()` sull'immagine della nostra maschera per rimuovere la filigrana. Ciò che fa realmente la funzione è che consente di rimuovere piccoli disturbi dall'immagine utilizzando un metodo chiamato `inpainting`.

La funzione può essere utilizzata con due algoritmi. Uno si basa sul metodo di marcia mentre l'altro si basa sulla meccanica dei fluidi. Puoi leggere di più su di loro e sulla funzione `cv2.inpaint()` [qui](#). Per fortuna, non dobbiamo scrivere quegli algoritmi da soli. Possiamo abilitare un oggetto flag all'interno della funzione `cv2.inpaint()` e scegliere qualsiasi algoritmo che vogliamo usare.

```
import cv2
import numpy as np
img=cv2.imread(path) # inserisci il percorso dell'immagine
hsv=cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
l_b=np.array([0, 0, 220])
u_b=np.array([255, 255, 255])
maschera=cv2.inRange(hsv, l_b, u_b)
dst = cv2.inpaint(img, maschera, 5, cv2.INPAINT_TELEA)
#dst = cv2.inpaint(img, maschera, 5, cv2.INPAINT_NS)
```

Puoi controllare i risultati dall'immagine sotto. La nostra funzione di ridipintura ha funzionato bene. Se pensi di dover ottenere risultati migliori, puoi provare a regolare i valori di soglia. È inoltre possibile utilizzare metodi come la sfocatura gaussiana per

migliorare la qualità dell'immagine della maschera.

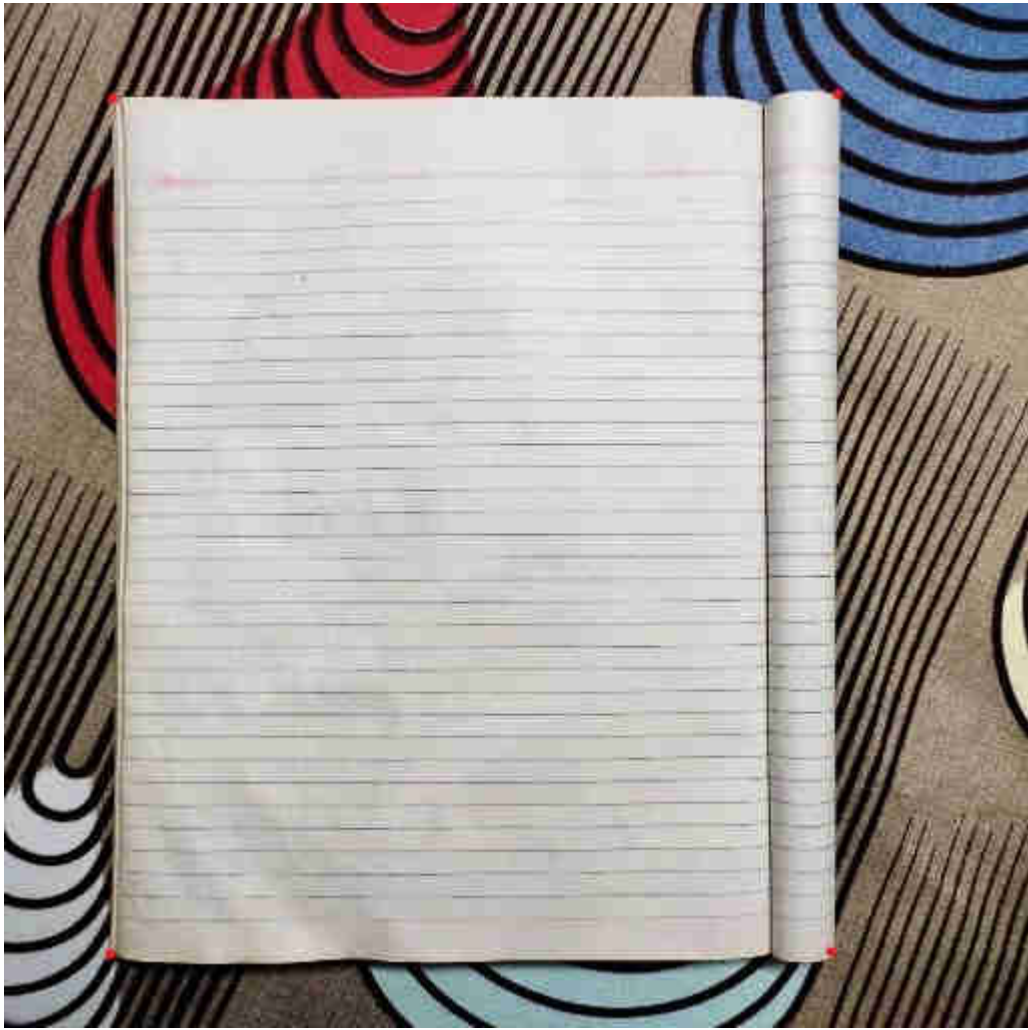


Rimuovere Watermark usando OpenCV, Python

## Rimozione dello sfondo utilizzando OpenCV e python

---

Quando si tratta di immagini, il rumore di fondo è molto comune. Utilizzando questa funzione di OpenCV, puoi estrarre solo informazioni utili dall'immagine e scartare tutto ciò che è presente in background. Una cosa buona è che ti permette di inserire la forma di output della tua immagine finale in modo da non dover indovinare quale sarà la dimensione della tua immagine. Per questo compito, utilizzerò l'immagine fornita.





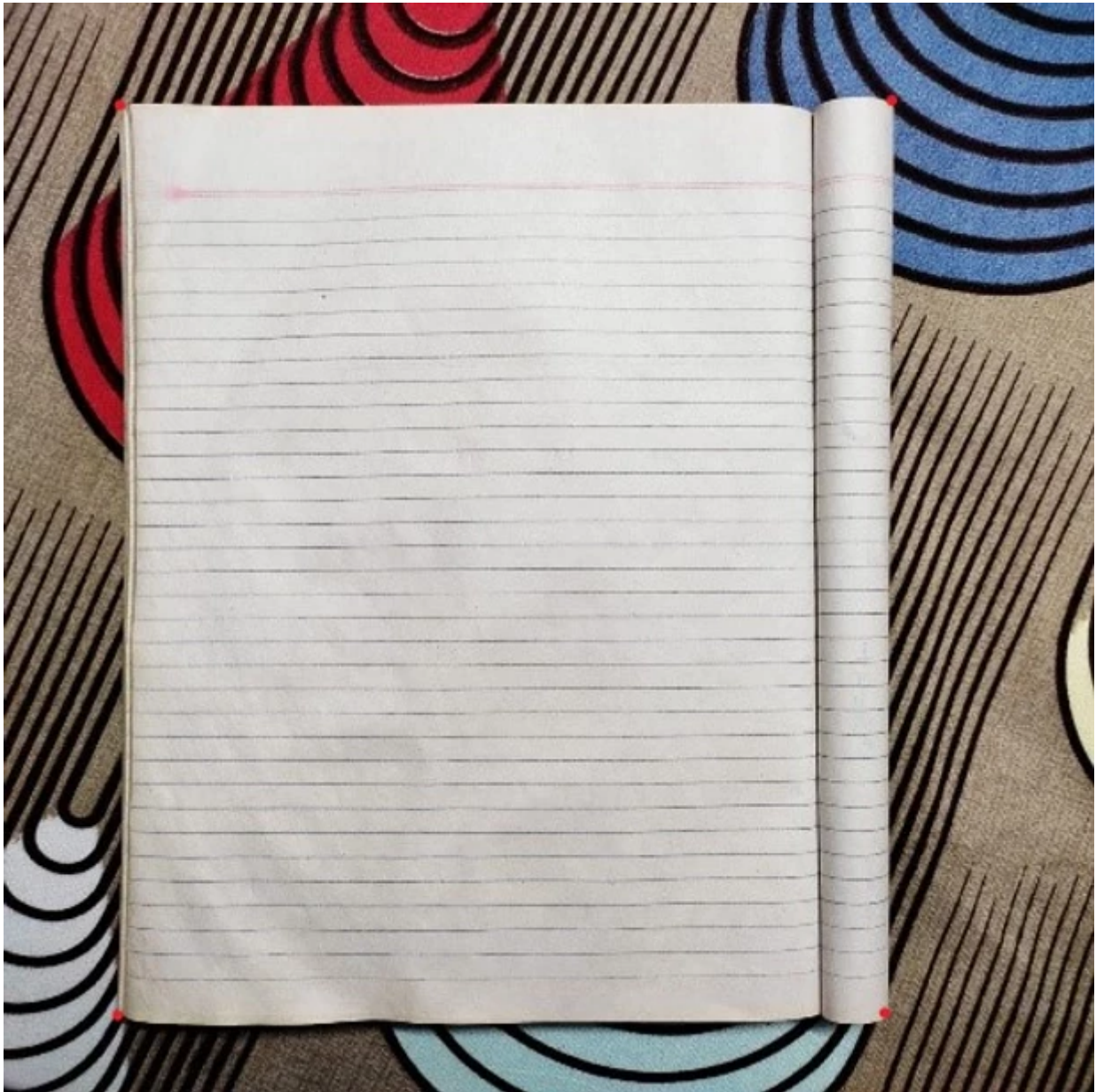
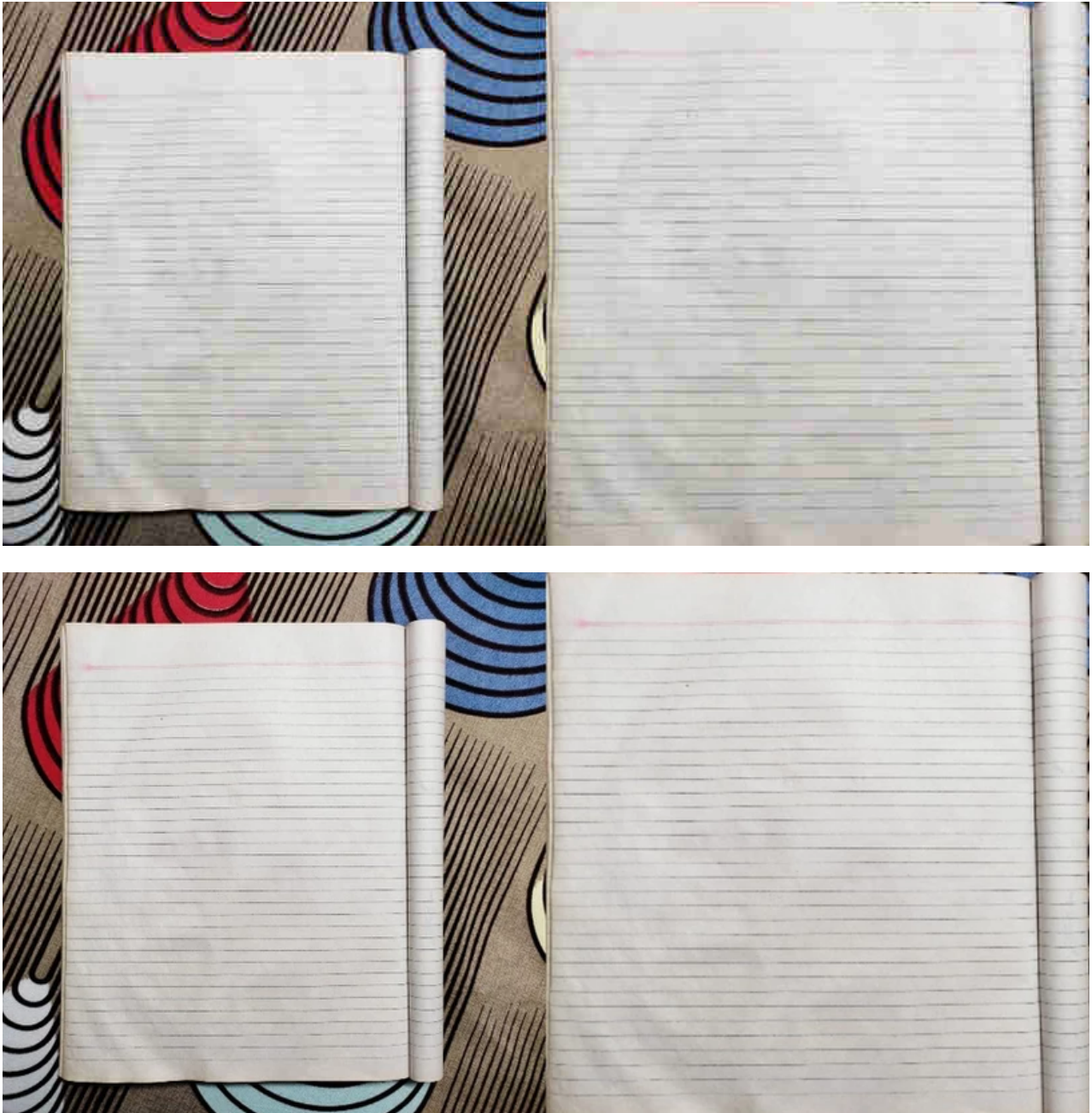


Immagine da cui estrarre solo il quaderno

Il nostro compito è rimuovere lo sfondo ed estrarre il primo piano (il taccuino) dall'immagine. La funzione che useremo richiede due insiemi di punti. Il primo set conterrà 4 punti che copriranno la regione in primo piano. Nell'immagine sopra, se guardi da vicino, sarai in grado di vedere 4 punti. Il primo set saranno le coordinate di questi punti. L'altro insieme di punti saranno i punti di confine della nostra immagine di output. Successivamente, useremo la funzione `cv2.getPerspective()` che ci darà la matrice di output e la funzione `cv2.warpPerspective` convertirà quella matrice nella forma desiderata.

```
import cv2
import numpy as np
img = cv2.imread(path) # inserisci il percorso dell'immagine
pts1 = np.float32([[57 , 49],[419 , 45],[414 , 477],[56 , 475]])
pts2 = np.float32([[0,0],[img.shape[0],0],[img.shape[0],img.shape[1]],
[0,img.shape[1]] ])
matrice = cv2.getPerspectiveTransform(pts1,pts2)
risultato = cv2.warpPerspective(img, matrice, (512,512))
cv2.imshow("risultato",risultato)
cv2.waitKey(0)
```



Risultato codice in python sopra descritto

**Crea filtri alle immagini usando OpenCV**

---

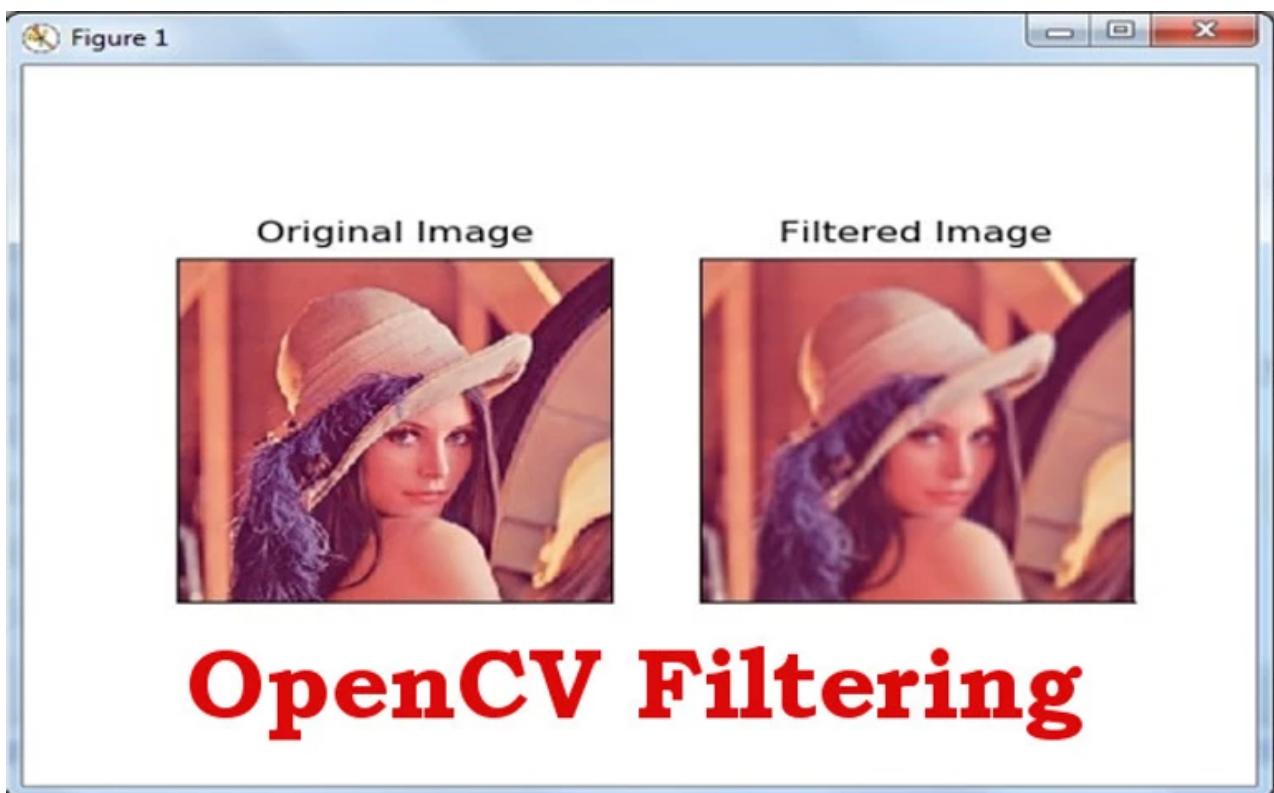
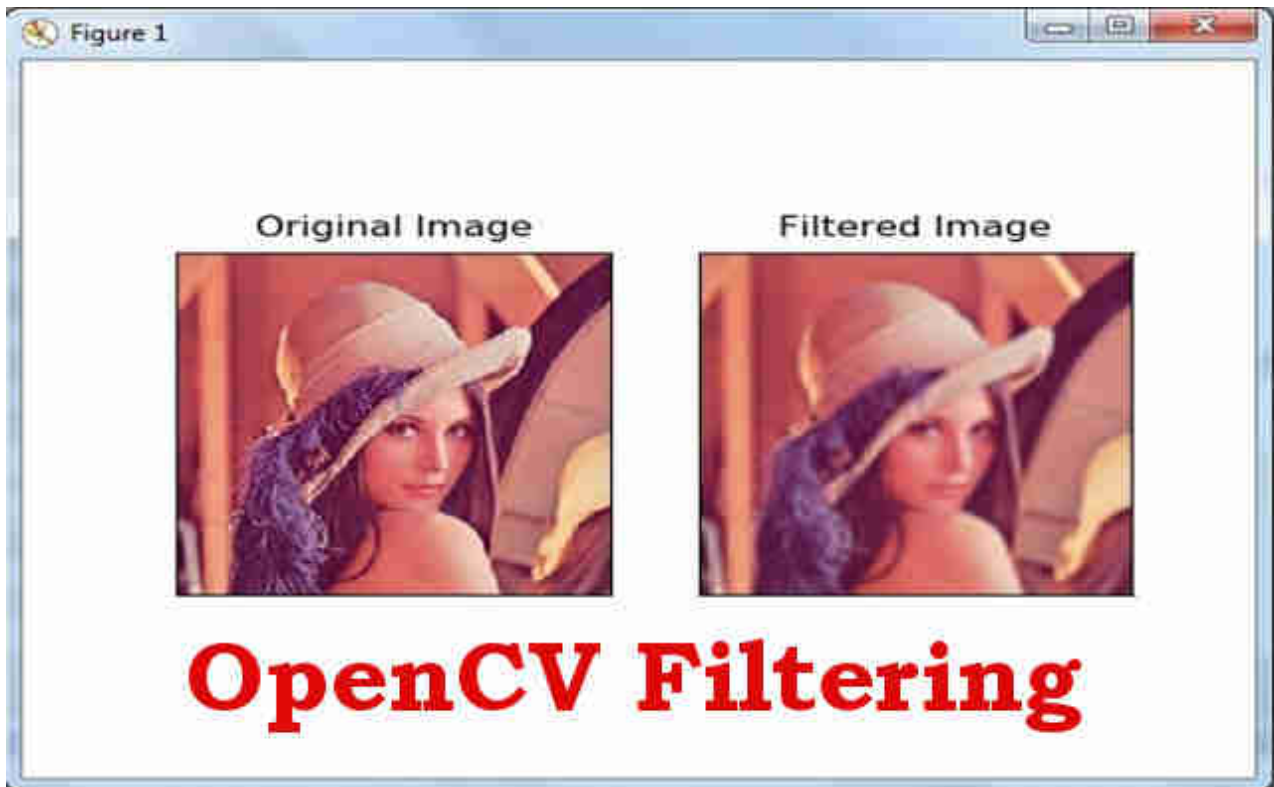
A differenza dei dati testuali, le immagini possono essere utilizzate per estrarre diverse funzionalità. Queste caratteristiche evidenziano le proprietà dell'immagine. Possiamo usare queste funzionalità per sfocare l'immagine, rendere più nitida l'immagine, estrarre i bordi dall'immagine e molto altro. Per eseguire questo compito utilizzeremo il concetto di convoluzione. La convoluzione è fondamentalmente il processo di applicazione del kernel su ogni pixel dell'immagine. nocciolo? Un kernel è una piccola matrice costituita da numeri che di solito si sommano all'unità. A seconda dei valori del kernel, possiamo estrarre diverse funzionalità da un'immagine.

In questo compito, andremo semplicemente a sfocare un'immagine usando le convoluzioni. Useremo il filtro sottostante (kernel) per il nostro compito. Questo kernel è anche chiamato filtro della media. Come suggerisce il nome, fa la media dei pixel circostanti e assegna il valore al pixel centrale. Puoi anche scegliere un diverso insieme di valori come preferisci, assicurati solo che tutti i valori si sommano a 1.

Dopo aver letto l'immagine, creeremo il nostro kernel e ne faremo un array NumPy. Quindi chiameremo la funzione `cv2.filter2d()`, che prenderà l'immagine di input, la profondità e il kernel. E questo è tutto. Possiamo semplicemente mostrare i nostri risultati.

```
import cv2
import numpy as np
img = cv2.imread(path) # inserisci il percorso dell'immagine
kernel = np.array([[1,1,1],
                  [1,1,1],
                  [1,1,1]])
kernel = kernel/9
res = cv2.filter2D(img, -1, kernel)
cv2.imshow("img",res)
cv2.waitKey(0)
```





Risultato codice in python sopra descritto

Puoi usare questo [link](#) per trovare filtri più sorprendenti che puoi applicare alle tue immagini preferite.

## Converti immagini in cartoni usando OpenCV

---

Tutti amano i cartoni animati e cosa c'è di meglio quando puoi crearne uno con la tua foto preferita. Con OpenCV, puoi svolgere facilmente questo compito utilizzando alcune delle funzioni di base. Fondamentalmente, l'intero processo include tre passaggi.

- Estrai i bordi dall'immagine.
- Applicare l'uniformità dell'immagine all'immagine in ingresso.
- Aggiungi i risultati dei passaggi 1 e 2.

Iniziamo leggendo la nostra immagine e convertendola in scala di grigi. Possiamo usare questa versione in scala di grigi dell'immagine e inviarla alla funzione `cv2.adaptiveThreshold` per ottenere i bordi dell'immagine. Puoi anche usare l'applicazione precedente per estrarre i bordi dall'immagine. Quindi dobbiamo applicare il livellamento alla nostra immagine. L'arrotondamento di un'immagine si riferisce alla creazione di una versione meno pixelata dell'immagine. In parole più semplici, stiamo solo riducendo il numero di pixel distinti nella nostra immagine. Ciò può essere ottenuto utilizzando la funzione di filtro bilaterale presente in OpenCV. Quindi possiamo semplicemente aggiungere l'immagine levigata e i bordi usando l'operatore `'bitwise_and'` in OpenCV per ottenere la nostra immagine del fumetto.

```
import cv2
import numpy as np
img = cv2.imread("images/lena.jpg")
grigio = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
grigio = cv2.GaussianBlur(grigio, (5,5), -1)
bordi = cv2.adaptiveThreshold(grigio, 255, cv2.ADAPTIVE_THRESH_MEAN_C,
cv2.THRESH_BINARY, 9,10)
color = cv2.bilateralFilter(img, 20, 245, 245)
cartoon = cv2.bitwise_and (color, color, mask = bordi)
cv2.imshow("cartone animato",cartoon)
cv2.waitKey(0)
```





Risultato codice in python sopra descritto

Puoi aggiungere più passaggi intermedi per migliorare i risultati come cambiare lo spazio colore, utilizzare filtri diversi, ecc.

## Conclusione

---

OpenCV è una parte essenziale per lavorare nella visione artificiale e utilizzandola possiamo creare migliaia di fantastiche applicazioni. Potresti aver pensato che alcune di queste applicazioni vengono utilizzate da noi nella vita di tutti i giorni. Qual è stata l'ultima volta che hai usato Cam Scanner per scansionare una copia dei tuoi compiti?

Pensi che la seconda applicazione possa aver avuto un ruolo in questo? L'immagine animata che abbiamo creato non sembra in qualche modo una versione Snapchat dell'immagine. Attraverso queste applicazioni, lo scopo era incoraggiarti a esplorare le soluzioni ai problemi della vita reale utilizzando strumenti semplici ed efficienti come OpenCV.

Le applicazioni che abbiamo trattato in questo articolo sono solo la punta dell'iceberg. Quando inizi a scavare più a fondo nel campo della visione artificiale, imparerai a conoscere concetti più interessanti e sorprendenti. Imparerai gli approcci di deep learning per risolvere problemi complessi. Imparerai a conoscere modelli pre-addestrati che potrebbero dare risultati eccellenti per un numero enorme di problemi. Addestrerai i tuoi modelli, creerai set di dati e troverai soluzioni a problemi che all'inizio sembrano irrealistici anche solo da chiedere.

Grazie. Spero che ti sia piaciuto leggere questo articolo. Condividilo ai tuoi colleghi o amici