

PROCESSAMENT D'IMATGES

Laboratori 1 - Report

curs 2023-24



Universitat
Pompeu Fabra
Barcelona

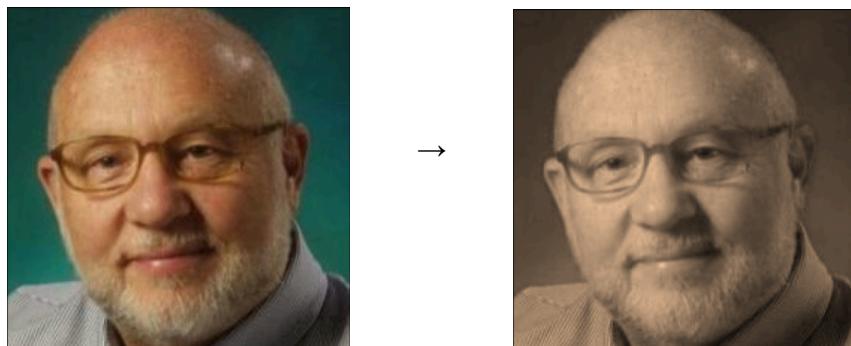
Xavier Riera - 268446
Pau Noguera - 265340

Exercice 1:

L'objectiu d'aquest exercici és crear un filtre de color sèpia per a imatges. Aquest filtratge es pot aconseguir mitjançant la modificació de l'espai HSV (*Hue-Saturation-Value*), és a dir, canviant els paràmetres dels components de l'espai de color.

Per aconseguir-ho, primer, s'ha de passar de l'espai RGB a l'espai HSV amb la funció “*rgb2HSV()*”. Després, en el nostre cas, només és necessari alterar el *hue* i la saturació, als valors 0.08 i 0.35 respectivament.

Finalment, es torna a convertir a l'espai RGB, mitjançant la funció “*HSV2RGB()*”, i aconseguim el següent resultat:

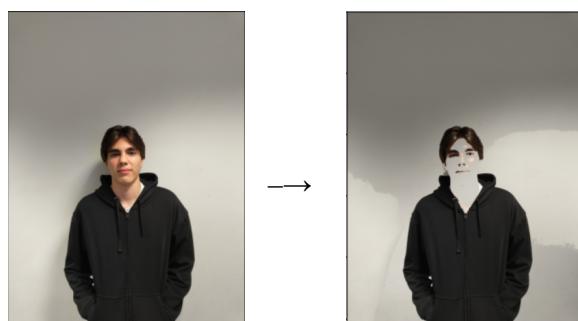


Exercice 2:

Pel següent exercici, hem decidit resoldre el programa de detecció de pell en imatges de les dues maneres possibles, és a dir, amb l'espai de colors RGB i el HSV.

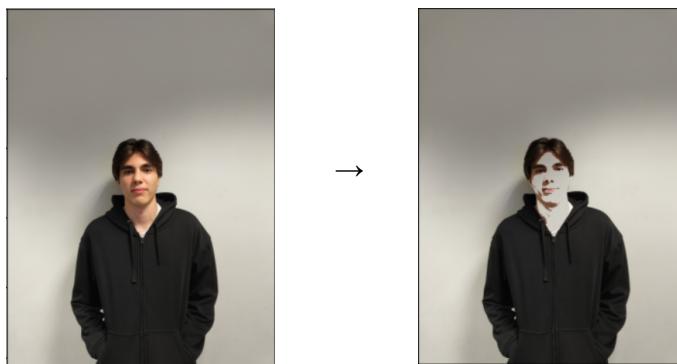
Pel primer cas, és necessari establir els tres canals (*red, green, blue*) i decidir uns intervals per a cada canal. Per fer-ho, necessitem generar les matrius amb respistes binàries (*True* o *False*) segons si superen o no el threshold triat. Llavors, per a que sigui un píxel de la pell, els tres canals han de coincidir en esser *True*.

Per tant, gràcies a dos bucles *for* niats, que recorren tots els píxels de la imatge. Per poder entrar dins l'*if* i substituir els píxels de la pell pels píxeles del fons, han de ser *True* (estar compresos dins l'interval) els píxel dels tres canals. Aconseguint el següent resultat:



Pel segon cas, es converteixen les dues imatges (retrato i fons) a l'espai de colors HSV amb la funció “*rgb2HSV()*”. I s'extreuen i guarden els components hue, saturació i value mitjançant la indexació de les dues imatges. I es comprova per cada píxel de la imatge retrato (amb l'iteració dels dos bucles for) si els valors del HSV es troben dins l'interval especificat pels píxels de la pell. Si compleixen el requisit, s'assignen els valors corresponents de la imatge de fondo a la imatge de retrato.

Finalment, es reconstrueix l'espai de colors HSV combinant els components modificats de matís, saturació i valor amb la funció “*np.vstack()*”. I es torna a convertir a l'espai RGB gràcies a la funció “*hsv2rgb()*”. Obtenint el següent resultat:



Basant-se en les observacions dels resultats obtinguts en els dos casos, podem concluir que:

En el cas de l'espai RGB, s'ha observat una transformació de la pell a la imatge de fons, excepte en els píxels més brillants de la pell. A més, s'ha notat una confusió significativa amb molts píxels de la paret, els quals també s'han interpretat com a pell. Per contra, en el cas de l'espai HSV, s'ha obtingut una imatge més precisa. Tot i això, s'ha observat que alguns píxels de pell més fosca no han estat interpretats com a pell, indicant una limitació en la detecció.

Així doncs, es pot concloure que el segon cas ha proporcionat un resultat més satisfactori, possiblement gràcies a la selecció d'un interval de color més adequat en l'espai HSV.

Exercice 3:

L'objectiu d'aquest exercici és generar una imatge a partir de la mitjana de dues imatges. Podem generar la nova imatge usant les dues operacions següents:

$$\frac{x+y}{2} \quad \text{o} \quad \frac{x}{2} + \frac{y}{2}$$

A més, per a calcular la nova imatge, podem realitzar els càlculs usant els valors double de cada pixel de les dues imatges originals o els valors en bytes de cada pixel de les imatges. Per tant, podem

obtenir quatre imatges noves segons la fórmula que apliquem i els valors de les dues imatges originals que usem. Les dues imatges que usarem són les següents:

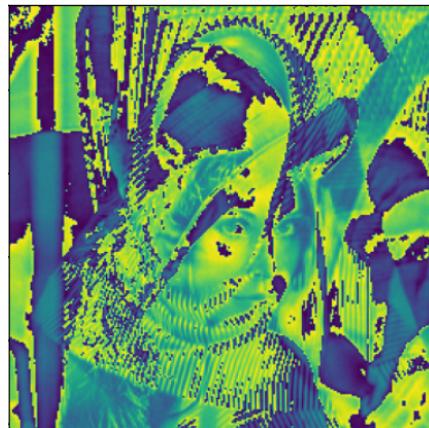


Lena



Barabara

En primer lloc, usarem la primera fórmula i els valors en bytes de les imatges. Abans d'aplicar la fórmula, transformarem les imatges a bytes mitjançant la funció `img_as_uint()` de la llibreria `skimage.util`. Aquesta funció transforma els valors de cada un dels píxels de la imatge en bytes. Una vegada hem passat a bytes les dues imatges, podem aplicar la fórmula i obtenir la següent imatge:



Aplificant la segona fórmula a les imatges en bytes obtenim la següent imatge:



A continuació, repetirem el procés i tornarem a aplicar les dues fòrmules però usant els valors double de les imatges. Tant si usam la primera fórmula com la segona obtenim la mateixa imatge:



Com podem veure, en tres dels quatre casos hem obtingut la mateixa imatge. L'únic cas en el que no hem obtingut la mateixa imatge és en el que hem aplicat la primera fórmula a les imatges en bytes. Aquest fet es degut a que quan es s'aplica la fórmula amb els valors en bytes de les imatges, els valors resultants s'ajusten al rang de valors d'un byte (8 bits, de 0 a 255). Per tant, quan s'aplica la primera fórmula i es sumen els valors de les dues imatges abans de fer la divisió, el valor supera el rang de valors d'un byte i el valor resultant es trunca o ajusta d'acord a aquest rang.