
COINT€R



L'ÚS DE LA VISIÓ ARTIFICIAL PER AL RECOMPTE DE MONEDES

Anna Pruença i Mascort

TREBALL DE RECERCA 2N DE BAT

TUTORA: YOLANDA FORT

INS de CELRÀ (Celrà)

CURS 2020-2021

DATA DE PRESENTACIÓ: 14 d'octubre de 2020

Resumen

La inteligencia artificial está dedicada al desarrollo de programas informáticos que intentan copiar la inteligencia humana, y cada vez está más presente en nuestras vidas. Una de las herramientas que usa es la visión artificial. Esta sirve para reconocer objetos, formas, patrones, de imágenes y extraer información a partir de distintos procesos.

Este trabajo pretende crear una herramienta útil para aquellas personas con dificultades para contar monedas por sí solas, ya sea por la edad o por otros motivos, con la visión artificial, ya que se observó que no existía ninguna con estas características en el mercado actual.

A partir de la investigación y el aprendizaje recibido a través de cursos online, se ha programado una aplicación, en lenguaje de Python, capaz de, a partir de una fotografía y la configuración de unos parámetros, reconocer distintas monedas y sumarlas.

Palabras clave: *visión artificial, Python, OpenCV, Kivy , fotografía, aplicación, identificación de monedas.*

Abstract

Artificial intelligence (AI) is dedicated to the development of computer programs which try to copy human intelligence, becoming really popular nowadays. One of the tools used for AI is called computer vision. This is used to recognize objects, shapes, patterns... from images and to extract information applying different procedures.

This work aims to create a useful tool for those people with difficulties to count coins on their own, either due to age or for other reasons, with computer vision, since it was observed that there were none with these characteristics in the current market.

Based on the research and learning received through online courses, an application has been programmed, in Python language, which is capable of, based on a photograph and the configuration of some parameters, recognize different coins and calculate the value of them all.

Key Words: *Computer Vision, Python, OpenCV, Kivy, photography, application, coin identifier.*

1 Índex

1	ÍNDEX	3
2	AGRAÏMENTS.....	5
3	INTRODUCCIÓ.....	6
3.1	PRESENTACIÓ	6
3.2	MOTIVACIÓ DEL TREBALL	6
3.3	OBJECTIUS DEL TREBALL.....	7
4	MÈTODE DE TREBALL.....	8
4.1	CERCA DOCUMENTAL	8
4.2	PART PRÀCTICA.....	8
5	PART TEÒRICA	9
5.1	QUÈ ÉS LA INTEL·LIGÈNCIA ARTIFICIAL?	9
5.2	QUI VA INVENTAR LA INTEL·LIGÈNCIA ARTIFICIAL?	9
5.3	VISIÓ ARTIFICIAL	9
5.3.1	<i>Adquisició d'una imatge.....</i>	<i>10</i>
5.3.2	<i>Processament d'imatges.....</i>	<i>12</i>
5.3.3	<i>Extracció d'informació</i>	<i>17</i>
5.4	PROGRAMARI.....	19
5.4.1	<i>Introducció</i>	<i>19</i>
5.4.2	<i>Màquina virtual</i>	<i>19</i>
5.4.3	<i>Sistema Operatiu</i>	<i>19</i>
5.4.4	<i>Llenguatges de programació</i>	<i>19</i>
5.4.5	<i>Llibreria</i>	<i>20</i>
5.4.6	<i>Editor.....</i>	<i>20</i>
6	PART PRÀCTICA	21
6.1	DESENVOLUPAMENT DE L'APLICACIÓ.....	21
6.1.1	<i>Etapa formativa en programació.....</i>	<i>21</i>
6.1.2	<i>Tria de programari</i>	<i>21</i>
6.1.3	<i>Instal·lació del programari.....</i>	<i>21</i>
6.1.4	<i>Procés del disseny de l'aplicació</i>	<i>22</i>
6.1.5	<i>Descripció de l'aplicació.....</i>	<i>25</i>
6.1.6	<i>Pantalla principal</i>	<i>26</i>
6.1.7	<i>Captura Càmera Web</i>	<i>26</i>
6.1.8	<i>Canals.....</i>	<i>27</i>
6.1.9	<i>Rotació.....</i>	<i>29</i>

6.1.10	<i>Desenfocament</i>	30
6.1.11	<i>Llindar</i>	32
6.1.12	<i>Erosió i dilatació</i>	33
6.1.13	<i>Enfocament</i>	33
6.1.14	<i>Costats</i>	34
6.1.15	<i>Monedes</i>	36
6.2	VERIFICACIÓ DEL FUNCIONAMENT DE L'APLICACIÓ	38
6.2.1	<i>Resultats</i>	39
6.2.2	<i>Accions de millora</i>	40
6.3	Usos	40
7	CONCLUSIONS	41
8	BIBLIOGRAFIA	43

2 Agraïments

Vull agrair l'ajut rebut durant aquest treball a les següents persones:

- A la meva tutora, la Yolanda Fort, per permetre'm treballar lliurement i donar-me el seu consell sempre que ho necessitava.
- A la Martina, per ajudar-me a l'hora de buscar un títol.
- A l'Òscar, per dedicar el seu temps a intentar compilar l'aplicació i a implementar el lector de veu.
- A la meva mare, per ajudar-me a ser constant en el treball i motivar-me i al meu pare, per ser la meva font d'energia i el meu suport emocional quan les coses no funcionaven.

3 Introducció

3.1 Presentació

En els darrers anys, la Intel·ligència Artificial (IA) ha anat agafant cada vegada més força, considerant-se un dels motors de la especulada IV Revolució industrial, permetent noves maneres en què la tecnologia s'insereix en les societats i fins i tot en el cos humà.

La IA, és cada vegada més present en el dia a dia, apareixent a una gran quantitat d'aplicacions i serveis, proporcionant eines que faciliten notablement el treball. Un dels sectors en la que s'utilitza seria l'automobilístic, on l'ús d'aquesta està relacionat amb el control i la detecció d'anomalies en el muntatge de peces de vehicles, aconseguint evitar futurs accidents deguts a aquests errors.

Aquest projecte intenta donar una solució a aquelles persones que presentin alguna limitació per comptar monedes, com per exemple la gent gran, aplicant una de les branques de la intel·ligència artificial, la visió artificial.

Aquest treball està estructurat en dues parts principals: una primera amb un estudi teòric, que s'ha basat en l'aprenentatge d'un llenguatge de programació i un recull d'informació sobre conceptes bàsics de la visió artificial, i una segona part, anomenada part pràctica, on s'exposa el disseny, la programació i els resultats d'una aplicació que reconeix i compta monedes.

3.2 Motivació del treball

Quan em disposava a triar un tema pel Treball de Recerca (TDR), em vaig trobar indecisa i sense saber què escollir. Volia fer un treball relacionat amb la construcció d'algun artefacte, ja que m'interessava bastant crear i dissenyar, però no sabia exactament què. En aquell temps, havia començat a fer un curs de programació i visió artificial, simplement pel plaer que m'aportava aprendre una cosa nova.

Quan vam assistir amb alguns alumnes de primer de batxillerat a una xerrada sobre bioinformàtica a Barcelona, se'm van obrir els ulls. Ens van dir que si volíem dedicar-nos a la investigació de qualsevol camp, en molts casos hauríem de saber programar. Com que jo portava temps pensant que em voldria dedicar a la investigació, vaig veure allà una possible idea de TDR que em servís per a millorar els meus coneixements sobre programació, i em permetés veure si realment volia cursar els estudis universitaris en aquest àmbit.

La idea de fer una aplicació per a comptar monedes em va venir un matí, quan em disposava a fer canvi d'unes 30 monedes a la meva mare per un bitllet de 10 euros. Durant el canvi, ella em va dir: «Com sé jo que no m'estàs enganyant?». Allà se'm va ocórrer la idea del projecte: dissenyar una aplicació per tal que aquest procés es pogués fer en qüestió de segons i sense haver de sortir de casa.

3.3 Objectius del treball

Es vol crear un sistema capaç de reconèixer i comptar monedes mitjançant un llenguatge de programació relativament senzill d'utilitzar però potent i un sistema de captació d'imatges.

- Objectiu principal:
 - Crear una aplicació per poder comptar monedes i el seu valor de forma automàtica, sense haver de sortir de casa.

- Objectius secundaris:
 - Programar una aplicació informàtica fàcil d'utilitzar i automàtica o manual, depenent de la preferència de l'usuari.
 - Conèixer el llenguatge de programació de visió artificial i saber-lo usar.
 - Aprofundir amb els coneixements sobre fotografia.
 - Aprendre noves tècniques de processament d'imatge.
 - Extreure informació d'una imatge.
 - Ajudar a les persones que no poden comptar monedes per elles mateixes.

4 Mètode de treball

4.1 Cerca documental

Primer s'ha fet una cerca per conèixer l'existència d'altres aplicacions similars per a dispositiu mòbil en el mercat actual, i s'han trobat varies aplicacions per comptar monedes: a partir del pes (*Coin Counter*), afegint manualment les unitats que es té de cada moneda (*CashCulator*) o algunes de reconeixement de moneda per a col·leccionistes (*Maktun: búsqueda de monedas por foto*); però no se n'ha trobat cap que utilitzi visió artificial per fer un recompte del valor total de monedes.

Un cop s'ha vist que aquesta aplicació no s'ha fet mai, es cercaran en quins àmbits s'utilitza la visió artificial, i en què consisteix aquesta. Després, es cercarà com s'obtenen les fotografies, i les eines de processament d'imatge que poden interessar pel projecte. També es buscaran les eines d'extracció d'informació necessàries per dur a terme l'aplicació.

4.2 Part pràctica

Primer es seleccionarà el programari necessari per poder fer l'aplicació. Un cop s'hagin trobat les opcions més adients pel treball, es procedirà a la programació. Es desenvoluparà l'aplicació amb el següent ordre: primer es programaran les eines de processament d'imatges i després es crearà la funció que identificarà les diferents monedes. Aquesta funció es té previst que funcioni basant-se en les proporcions entre monedes.

Un cop l'aplicació estigui finalitzada, es faran proves per tal de comprovar la seva funcionalitat, i amb quins paràmetres funciona millor.

5 Part teòrica

5.1 Què és la intel·ligència artificial?

La intel·ligència artificial (IA) és la intel·ligència portada a terme per màquines. En termes de computació, una màquina “intel·ligent” ideal és un agent flexible que percep el seu entorn i realitza accions que maximitzen les seves possibilitats d’èxit amb algun objectiu.

És considerada una part de la informàtica, la qual es dedica al desenvolupament d’algoritmes que permeten a una màquina prendre decisions intel·ligents o comportar-se com si tingués una intel·ligència semblant a la humana.

5.2 Qui va inventar la intel·ligència artificial?

El britànic Alan Turing es considera el creador de la intel·ligència artificial, ja que al llarg de la Segona Guerra Mundial, mentre treballava amb la intel·ligència britànica, va començar a fonamentar les bases del que avui coneixem com a intel·ligència artificial, dissenyant una màquina capaç de realitzar qualsevol càlcul prèviament definit.

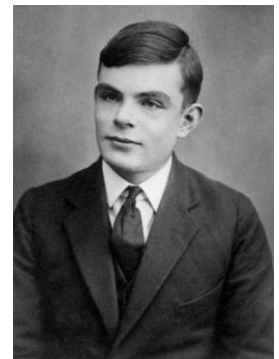


Figura 1. Alan Turing. Font: https://es.wikipedia.org/wiki/Alan_Turing

Una altra persona que va contribuir en el desenvolupament de la IA és Artur L. Samuel, que al 1956 va crear una màquina capaç de guanyar-lo jugant a les dames, un joc simple però estratègic. Per poder aconseguir-ho, havia d’ensenyar a la màquina l’estratègia per guanyar partides, i no només les bases dels moviments de les fitxes.



Figura 2. Artur L. Samuel amb la seva màquina de jugar a dames. Font: <https://www.ibm.com/ibm/history/ibm100/us/en/icons/ibm700series/impacts/>

5.3 Visió artificial

La visió artificial o visió tècnica és un camp de la intel·ligència artificial. La seva finalitat és permetre que un ordinador pugui analitzar unes característiques d’una imatge i pugui treure’n unes conclusions. Per a aconseguir-ho es segueix un procés format per tres fases:

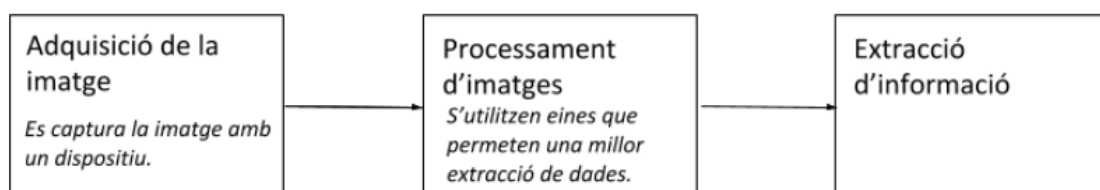


Figura 3 Procés que segueix l'ordinador al utilitzar la intel·ligència artificial. Font: Imatge de l'autor (30-08-20)

Les aplicacions pràctiques de la visió artificial són:

- Detectar, segmentar, localitzar i reconèixer certs objectes en una imatge (exemple: Detecció de defectes en alguns lots, com podrien ser ratllades en metalls, errors de pintura, males impressions,...)
- Registrar diferents imatges d'un mateix objecte o escena i fer concordar un mateix objecte en diverses imatges (exemple: *reconeixement facial del mòbil*).
- Fer un seguiment d'un objecte en una seqüència d'imatges (exemple: Seguiment de persones a través del circuit de càmeres d'una ciutat).
- Fer una estimació de les imatges tridimensionals d'humans. (exemple: Interpretació d'una Tomografia Axial Computada (TAC) mèdic.)
- Cercar contingut a través d'una imatge (exemple: *Google Lens*)

Per poder aconseguir aquests objectius, s'apliquen reconeixement de patrons, aprenentatge estadístic i processament d'imatges, entre d'altres.

5.3.1 Adquisició d'una imatge

5.3.1.1 Llum visible

La llum és constituïda per radiacions electromagnètiques de diferents longituds d'ona , però per a l'ull humà només li és possible captar les radiacions que tenen una longitud d'ona que va dels 400 nm (violeta) als 700 nm (vermell), aquest interval és conegut com a l'espectre visible.

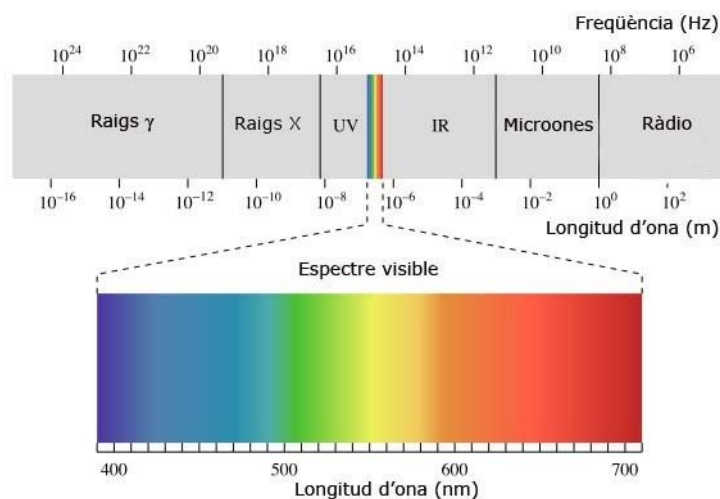


Figura 4 . Representació de l'espectre electromagnètic. Font: <https://www.ibm.com/ibm/history/ibm100/us/en/icons/ibm700series/impacts/>

5.3.1.2 Captura d'imatges digitals

Al parlar de sensors, les persones tendeixen a pensar en les càmeres, ometent el sensor propi del cos humà, els ulls. Aquests capturen les imatges a la retina, les cèl·lules (cons i bastons) de la qual perceben els colors, i generen en impulsos elèctrics que són enviats al cervell que els transforma en imatges.

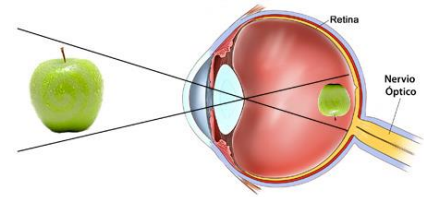


Figura 5. Representació formació d'una imatge a la retina.

Font: <https://sites.google.com/site/elcoloriofite123/home/1-el-color-com-a-fenomen-fisic>

Les càmeres fotogràfiques utilitzen el mateix sistema de captació d'imatges que l'ull humà, amb la diferència que en comptes de tenir cèl·lules, tenen petites cavitats sensibles a la llum, també anomenades *fotosites*. Quan es prem el disparador, aquestes cavitats són destapades per tal de captar fotons¹. Quan s'acaba el temps d'exposició, la càmera tanca els *fotosites* i genera una senyal elèctrica proporcional als fotons capturats.

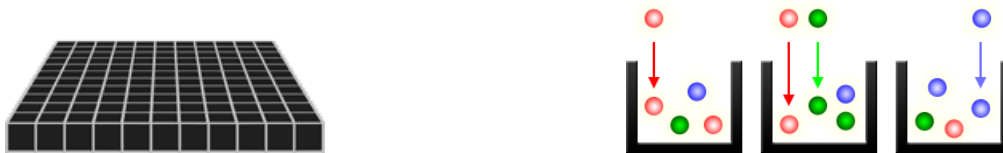


Figura 6. Representació sensor blanc i negre. Font: <https://www.cambridgeincolour.com/tutorials/camera-sensors.htm>

Per crear imatges amb color, s'ha de poder saber quina quantitat hi ha de cada color. Per fer-ho, s'aplica un filtre a sobre les cavitats que només permet passar un color determinat (vermell, blau o verd).

Un dels filtres més utilitzats per a capturar imatges és el "Bayer Array" (veure figura 7). Aquest filtre assimila cada quatre cavitats com un punt de llum o píxel de la imatge, barrejant el verd, blau i vermell de cada apartat creant un color. Aquest és el sistema de representació del color RGB (Red Green and Blue).

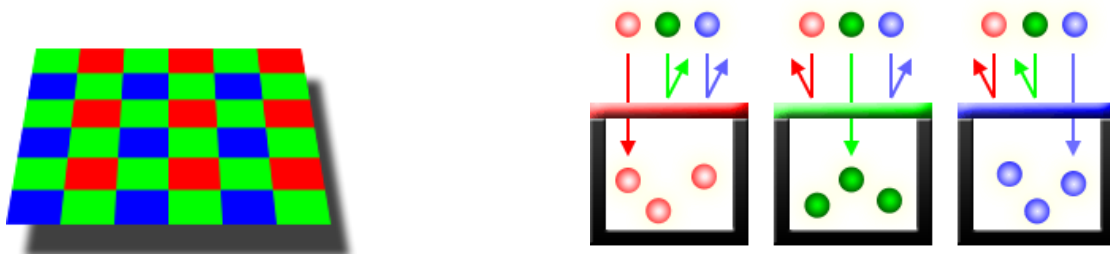


Figura 7. Representació del sensor d'imatge a color. Font: <https://www.cambridgeincolour.com/tutorials/camera-sensors.htm>

¹Amb física moderna, el fotó és la partícula elemental responsable de les manifestacions quàntiques del fenomen electromagnètic. És la partícula que porta totes les formes de radiació electromagnètica.

A la figura 8, hi ha el doble de cavitats verdes que de vermelles i blaves. Això és perquè l'ull humà és més sensible al color verd.

La imatge real capturada es converteix així en informació digital que podrà ser processada.



Figura 8. Representació filtre del sensor "Bayer Array". Font: <https://www.cambridgeincolour.com/tutorials/camera-sensors.htm>

5.3.2 Processament d'imatges

El processament d'imatges és el conjunt de tècniques digitals que s'apliquen a les imatges un cop digitalitzades amb l'objectiu de millorar la qualitat o facilitar la recerca d'informació.

5.3.2.1 Selecció espai de color de treball

L'espai de color és un sistema de representació del color. N'hi ha de diferents tipus:

- **RGB**

Tal com s'ha comentat a l'apartat 5.3.1.2, les imatges són capturades en espai RGB. La imatge està composta per 3 matrius superposades, cada una de les quals representa la informació de color d'un canal: vermell, blau o verd (veure figura 9).

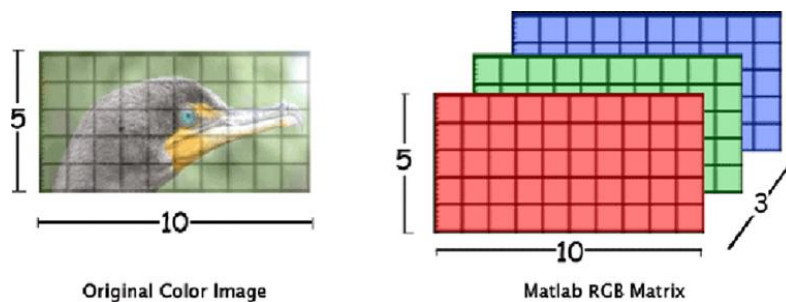


Figura 9. Representació d'una imatge i de les 3 matrius RGB.
Font: https://www.researchgate.net/figure/Color-image-representation-and-RGB-matrix_fig15_282798184

- **Blanc i negre**

La informació és transformada en una matriu d'una sola dimensió, on el blanc val 255 i el negre 0, i tota la resta de valors oscil·len entre aquests dos extrems, creant una escala de grisos.

- **Model HSV (Hue, Saturation, Value - tonalitat, saturació, valor)**

Es tracta d'una transformació no lineal² de l'espai de color RGB, i és utilitzat en progressions de color.

-Tonalitat (H): és el tipus de color (blau, verd, vermell,...). És representat com a un cercle, on cada valor equival a un color.

-Saturació (S): Els valors poden anar del 0 al 100. Com menys saturació tingui un color, més tonalitat grisenca tindrà i més descolorit estarà.

-Valor (V): equival a la brillantor del color. Els valors possibles van del 0 al 100%. Com en els altres casos, 0 és negre, però depenent de la saturació, 100 podria ser blanc o un color més o menys saturat.

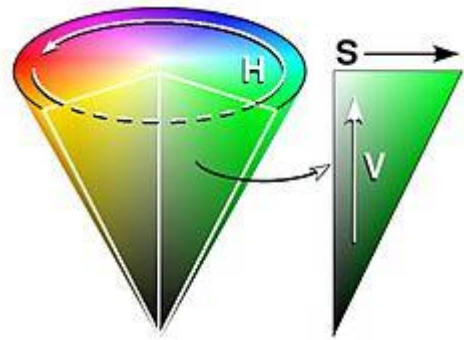


Figura 10. Representació espai de color HSV. Font: https://es.wikipedia.org/wiki/Modelo_de_color_HSV



Figura 11. Imatge amb vermell.
Font: Imatge de l'autor (20-11-2019)



Figura 12. Imatge amb filtre blanc i negre.
Font: Imatge de l'autor (20-11-2019)

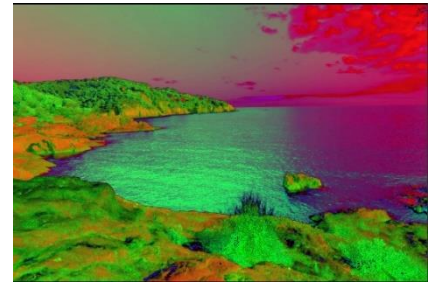


Figura 13. Imatge amb HSV.
Font: Imatge de l'autor (20-11-2019)

5.3.2.2 Escala i rotació

L'escala i la rotació formen part del grup de transformacions geomètriques d'una imatge, és a dir, de la modificació de coordenades d'una imatge.

Escala: És utilitzat per a poder canviar la mida d'una imatge.

Rotació: Permet canviar l'orientació de la fotografia.



Figura 14. Imatge amb una escala 0.36.
Font: Imatge de l'autor (20-11-2019)



Figura 15. Imatge amb una rotació amb angle -149.
Font: Imatge de l'autor (20-11-2019)

² Una transformació no lineal és aquella que no equival a una recta.

5.3.2.3 Erosió i dilatació

L'erosió i la dilatació formen part del grup de transformacions morfològiques, que són operacions simples basades en la forma de la imatge. Aquests processos solen ser utilitzats per a l'eliminació de soroll³.

Erosió: En el procés d'erosió, s'aplica un kernel⁴ a la imatge. Al aplicar la operació d'erosió, s'obté un efecte d'aprimament de l'àrea que compon la imatge.



Figura 16. Platja de Tamariu.
Font: Imatge de l'autor (20-11-2019)



Figura 17. Imatge on se li ha aplicat una erosió 8 vegades. Font: Imatge de l'autor (20-11-2019)

Dilatació: El procés de dilatació és totalment oposat al de l'erosió. Al aplicar l'operació sobre una imatge, s'obté l'efecte d'expansió o ampliació de la regió en la que s'estigui treballant.



Figura 18. Platja de Tamariu.
Font: Imatge de l'autor (20-11-2019)

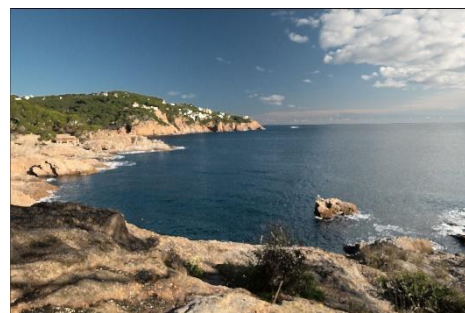


Figura 19. Imatge on se li ha aplicat 8 vegades una dilatació.
Font: Imatge de l'autor (20-11-2019)

5.3.2.4 Desenfocament

El desenfocament també forma part del grup de transformacions morfològiques. Consisteix en realitzar una transformació de forma que hi hagi menys diferències entre els valors dels colors dels píxels d'una imatge, la qual cosa tendeix a suavitzar la imatge. L'impacte de la transformació dependrà del model de kernel utilitzat: en alguns tots els punts tenen la mateixa influència, en canvi en d'altres hi ha prioritats.

³ El soroll són dades sense significat, que no s'estan utilitzant per a transmetre un senyal, sinó que són simplement un resultat no desitjat d'altres activitats que se sumen amb els senyals que que es volen rebre, la qual cosa modifica les dades i pot crear interferències.

⁴ Un kernel és la matriu de transformació responsable de generar l'efecte.

Els tres models de kernel de desenfocament utilitzats en el treball són:

-Median: Tots els punts tenen la mateixa importància. Es genera un desenfocament igual a tota la imatge.



Figura 20. Imatge amb filtre de desenfocament Median 11 i filtre de desenfocament Median 11.
Font: Imatge de l'autor (20-11-2019)

-Gauss: S'observen els punts del costat de cada píxel per a determinar el seu valor. Com més a prop del píxel en qüestió, més importància tindrà el seu valor. Es genera un desenfocament general però és més nítid que el Median.



Figura 21. Imatge amb filtre de desenfocament Gauss 11 i filtre de desenfocament Gauss 11.
Font: Imatge de l'autor (20-11-2019)

-Bilateral: S'aplica un desenfocament Gauss, però si dos punts tenen una diferència de color molt gran, es percep com a contorn i no es té en compte aquell punt. Es genera un desenfocament general però es veuen clarament els contorns.

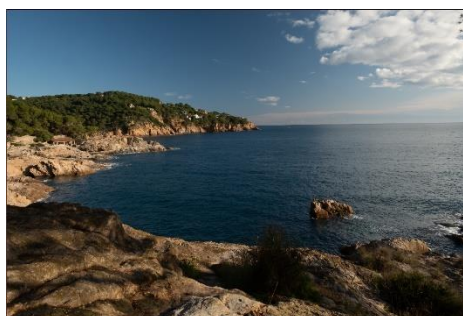


Figura 22. Imatge amb filtre de desenfocament Bilateral: radi 9, sigma c 75, sigma x 75.
Font: Imatge de l'autor (20-11-2019)

5.3.2.5 Enfocament

L'enfocament forma part del grup de transformacions morfològiques. L'enfocament consisteix en una transformació inversa a la del desenfocament, ja que es tracta de que hi hagi una diferencia de colors major entre els píxels d'una imatge, generant més contrast.

L'impacte dependrà del model de kernel utilitzat

- Enfocament: genera contrast. (Veure figura 23)
- Pas alt: genera molt de contrast. (Veure figura 24).



Figura 23. Imatge amb enfocament.
Font: Imatge de l'autor (20-11-2019)



Figura 24. Imatge amb enfocament pas alt.
Font: Imatge de l'autor (20-11-2019)

5.3.2.6 Llindar

El llindar és una transformació matemàtica que a partir d'un valor determinat manualment, permet transformar la informació del píxel en 0 (negre) o 255 (blanc). Si el valor del píxel és més gran que el valor de el llindar, se li assigna 255 i si és inferior, se li assigna 0. El resultat és una imatge on els píxels són en blanc o negre.

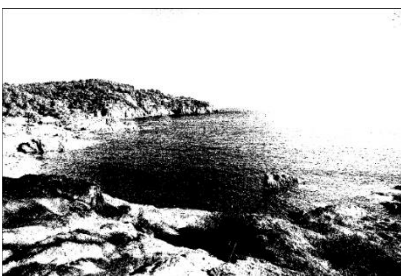


Figura 25. Imatge amb llindar de 57.
Font: Imatge de l'autor (20-11-2019)



Figura 26. Imatge amb llindar de 107.
Font: Imatge de l'autor (20-11-2019)



Figura 27. Imatge amb llindar de 165.
Font: Imatge de l'autor (20-11-2019)

5.3.3 Extracció d'informació

5.3.3.1 Detecció de contorns

Un contorn és el canvi sobtat en la diferència de color entre dos píxels d'una imatge. Aquesta propietat permet identificar formes i objectes de la fotografia.

Totes aquestes eines treballen en imatges amb blanc o negre.

Sobel: Es focalitza en els eixos verticals i/o horitzontals. No extreu els costats com fa el Canny o el Laplace, sinó que calcula la diferència de color entre els punts consecutius. És una representació del gradient de color en la direcció X i/o Y.

Canny: Extreu els costats basant-se en la diferència de color entre els píxels adjacents. Quan detecta que hi ha dos píxels que tenen una diferència igual o més gran al límit superior de detecció de costats, que s'estableix manualment, al píxel se li atorga un valor de 255 (blanc). En el cas contrari, el píxel queda amb un valor de 0 (negre). D'aquesta forma, a diferència de Sobel que representa tots els gradients⁵, només extreu la informació rellevant.

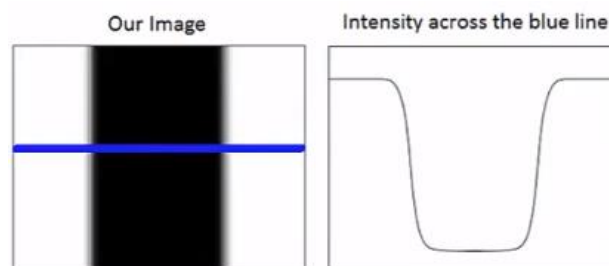


Figura 28. Representació de la intensitat dels punts del costat de la línia blava. Font: <https://www.udemy.com/course/master-computer-vision-with-opencv-in-python/learn/lecture/5860508#overview>

En aquesta funció també es pot inserir un límit inferior de detecció de costats, que s'utilitza pels punts que tenen un valor de gradient menor al límit superior de detecció de costats, però que estan connectats a algun píxel que sí que el superi. D'aquesta manera formen part del contorn alguns punts que no tenien un gradient tant gran però que formaven part de l'objecte.

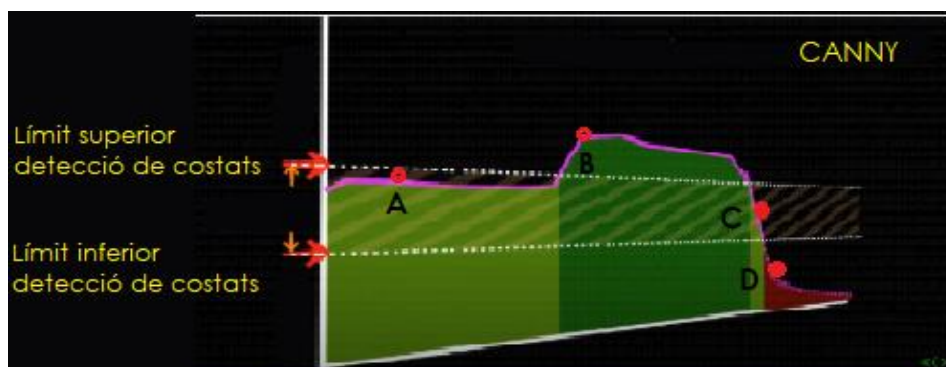


Figura 29. Representació del funcionament de la detecció del gradient del filtre Canny. Font: <https://www.youtube.com/watch?v=sRFM5IEqR2w>

⁵ Diferència de color entre dos punts.

La figura 29 és una representació de tots els punts del contorn d'una imatge, en el que la línia violeta representa els valors de gradient de color dels píxels. Com es pot observar en els punts A i C, són acceptats com a contorn degut a que estan connectats al punt B, que té un valor més gran al límit superior de detecció de costats, i tenen un valor per sobre el límit inferior de detecció de costats. En canvi, el punt D, no és considerat contorn perquè, per molt que estigui connectat al punt B, no té un valor que superi al límit inferior de detecció de costats.

Laplace: Laplace extreu els costats aplicant la segona derivada del color a tots els punts de l'imatge. Els punts que la seva segona derivada corresponen amb el valor 0, se'ls hi assigna un valor de 255 (blanc). Els punts on la segona derivada és zero, són els punts en els que la primera derivada o gradient de color, es màxim. Com s'ha explicat anteriorment, si el gradient és màxim significa que forma part d'un contorn.

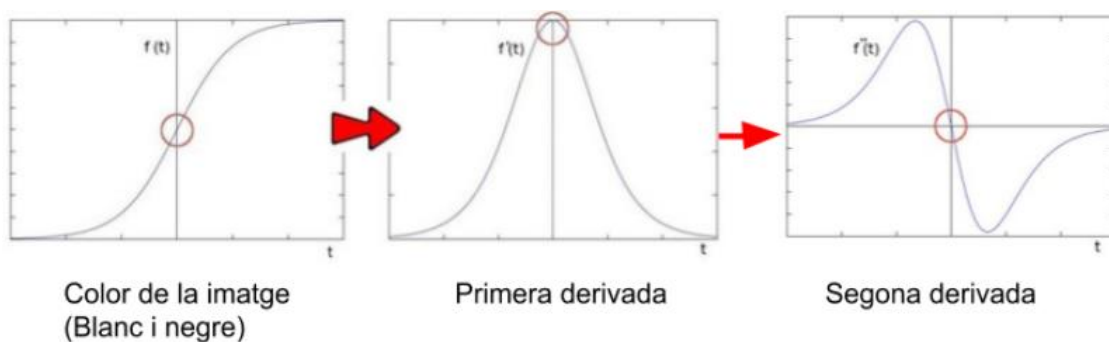


Figura 30. Derivada el color respecte el punt del costat de la imatge.

Font: https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/laplace_operator/laplace_operator.html

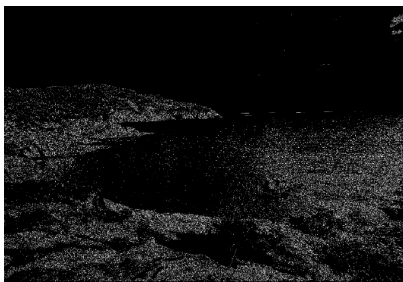


Figura 31. Imatge amb Canny amb el gradient superior a 127 i el mínim a 75 .
Font: Imatge de l'autor (8-11-2019)

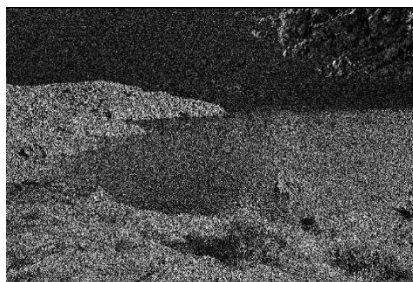


Figura 32. Imatge amb Sobel, on la matriu és de 5 i està en direcció y.
Font: Imatge de l'autor (8-11-2019)



Figura 33. Imatge amb Laplace, on la matriu és de 7.
Font: Imatge de l'autor (8-11-2019)

5.3.3.2 Càlcul d'àrees

L'àrea es determina com el càlcul de la superfície delimitada pel contorn.

5.3.3.3 Càlcul del centre del polígon

El centre del polígon es determina com el centre geomètric de la superfície delimitada pel contorn.

5.4 Programari

5.4.1 Introducció

A continuació es mostra un esquema amb les diferents parts del programari que posteriorment s'explicaran.

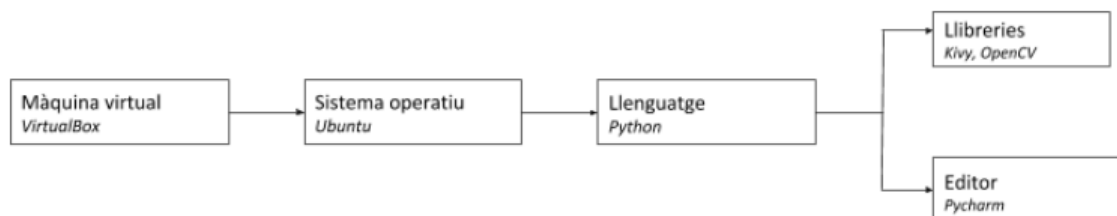


Figura 34. Esquema de les parts del programari. Font: Imatge de l'autor (14-8-20)

5.4.2 Màquina virtual

Una màquina virtual és un programari que imita un ordinador, i és capaç d'executar programes com si fos un ordinador real, malgrat ser virtual. Això permet que els usuaris puguin instal·lar dins la màquina virtual un sistema operatiu diferent de la màquina real. Aquesta màquina virtual és independent de la instal·lació física i per tant pot ser executada en ordinador diferents. Una característica essencial d'aquestes és que els processos que s'executen en aquesta no poden escapar de l'ordinador virtual.

La Virtual Box és un tipus de màquina virtual Open Source utilitzada tant per empreses com per ús particular.

5.4.3 Sistema Operatiu

El sistema operatiu és el conjunt dels diferents programes que controlen el funcionament d'un ordinador. Sobre aquest sistema s'executen les aplicacions de l'usuari.

Hi ha diferents tipus de sistemes operatius, un dels Open Source és l'Ubuntu.

5.4.4 Llenguatges de programació

Un llenguatge de programació és un llenguatge informàtic utilitzat per controlar el comportament d'una màquina, normalment un ordinador. Aquest, té diferents regles sintàctiques i semàntiques que s'han de seguir per poder crear un programa informàtic. Hi ha molts tipus de llenguatges de programació, i una manera de classificar-los és amb la Classificació alt nivell-baix nivell. Els llenguatges de baix nivell són els que s'assemblen més al codi binari, mentre que els llenguatges d'alt nivell són els que s'assemblen més al llenguatge humà.

Un dels llenguatges de programació a destacar és el Python. Aquest és un llenguatge de programació d'alt nivell molt utilitzat. El seu creador va ser Guido Van Rossum el 1991. La particularitat del disseny de Python és que aquest busca llegibilitat en el codi, i que gràcies a la seva sintaxi els programadors puguin expressar conceptes en menys línies de codi que altres llenguatges de programació. És Open Source.

5.4.5 Llibreria

Una llibreria és la sèrie de funcionalitats en un llenguatge de programació específic. Es criden des dels programes per a realitzar tasques concretes, simplificant les tasques de programació ja que no s'han de crear de nou. No consisteixen en un programa en sí mateixes.

5.4.5.1 Kivy

Kivy és una llibreria de Python Open Source, que permet desenvolupar programes amb una interfase tipus finestra.

5.4.5.2 OpenCV

OpenCV és una llibreria Open Source de visió artificial desenvolupada per Intel. És una de les llibreries de visió artificial més popular en l'actualitat. Permet detectar moviment, reconèixer objectes, reconstrucció 3D a partir d'imatges,...

5.4.6 Editor

Un editor o IDE (acrònim en anglès de Integrated Development Environment) és una eina informàtica per permetre un desenvolupament de programari còmode i ràpid. Un dels propòsits dels IDE és reduir la configuració necessària per reconstruir múltiples utilitats de desenvolupament, aconseguint una major productivitat de desenvolupament, perquè és més fàcil aprendre a fer servir un IDE que integrar manualment totes les eines per separat.

El PyCharm és un IDE, utilitzat en la programació del llenguatge Python. La versió Community és d'ús gratuït.

6 Part pràctica

6.1 Desenvolupament de l'aplicació

6.1.1 Etapa formativa en programació

Per poder començar a elaborar aquest projecte, s'ha hagut d'aprendre els coneixements bàsics de Python i les llibreries OpenCV i Kivy a través de dos cursos de la plataforma Udemy: *"Python para Android, iOS, Windows, Linux y Mac - Desde cero"* i *"Master Computer Vision™ OpenCV4 in Python with Deep Learning"*. Paral·lelament, s'han consultat tutorials a Youtube per poder consolidar aquests coneixements i complementar-los.

6.1.2 Tria de programari

S'ha fet una cerca documental per poder comprovar si les diferents eines que s'explicaven durant el curs eren o no les opcions idònies per a realitzar el treball.

Primer s'ha investigat que el llenguatge Python fos suficient. Per fer-ho, s'han cercat els més populars i utilitats actualment, i s'ha observat que són: Java, C, Python, C++, C# i Visual Basic.NET. Per a determinar si Python era l'adient, s'ha basat en dos factors: que fos Open Source i que fos senzilles d'utilitzar per principiants en la programació. S'ha comprovat que totes eren Open Source, per tant, el Python no ha destacat amb això. Després, s'ha buscat quin era el grau de dificultat de cada llenguatge, i s'ha trobat que el Python era l'ideal per a principiants, degut a la seva senzillesa i similitud al llenguatge anglès.

En el curs que s'estava seguint de Python s'utilitzava el PyCharm com a editor, i per tal de garantir que s'estava prenent una bona decisió, s'han cercat els més populars per a fer-ne una comparació. S'ha trobat que PyCharm és el segon més usat, per darrera de PyDev per a Eclipse, i s'ha considerat que és un programa reconegut pel mercat, i que era perfectament vàlid per al projecte.

Finalment, s'ha comprovat la validesa de les llibreries Kivy i OpenCV. Se n'han buscat d'altres Open Souce, per comparar. S'ha vist que per a visió artificial, la llibreria més utilitzada és OpenCV, al tenir més de 500 algoritmes per a processament d'imatge, descripcions geomètriques, segmentació,... També s'han buscat llibreries que permetessin crear una interacció fàcil amb l'usuari i s'ha vist que Kivy era una de les més utilitzades i senzilles. Per tant, s'ha mantingut la tria inicial.

6.1.3 Instal·lació del programari

Per poder aconseguir utilitzar Python sense haver de modificar tot el software de l'ordinador, s'ha instal·lat una màquina virtual. A dins d'aquesta màquina virtual ha estat necessari instal·lar tot el programari de Python, Pycharm, Kivy i OpenCV. A l'annex s'adjunta el manual creat per l'autor d'aquest treball utilitzat per instal·lar tot el programari necessari, ja que no s'ha recollit l'informació d'un sol lloc, sinó que s'ha recopilat de varies fonts diferents.

6.1.4 Procés del disseny de l'aplicació

S'ha seguit el següent procés per tal de dissenyar la aplicació:

Primer es generen pantalles amb *widgets* des de Python: s'observa que té un grau de dificultat molt elevat perquè es necessiten pantalles amb molts botons. Aquesta finalment es programa amb Kivy ja que permet construir pantalles més completes de manera més simple.

Es programen els paràmetres següents: pantalla inicial, canals, rotació, desenfocament, lllindar, erosió i dilatació, enfocament, costats i càmera. (veure apartat 7.1.5 on es detalla cada un d'aquests).

Un cop acabada la programació dels paràmetres mencionats, es generen les dades mestres de proporcions de les monedes (veure figura 34). S'observa un problema greu que fa plantejar-se la viabilitat del projecte: hi ha proporcions que entre elles tenen una diferència més petita del 0,3%. Al tractar-se de fotografies fetes manualment, s'ha de contemplar que depenent del grau d'enfocament, la llum, la lent, les ombres,... es poden generar distorsions a la forma de la moneda, per la qual cosa es necessita un marge d'error per tal d'evitar una detecció incorrecta del valor de la moneda. S'analitza la possibilitat de afegir paràmetres com el color posteriorment, però primer es decideix experimentar amb monedes de 2€, 1€ i 50 ct, ja que la diferència entre les seves proporcions és més gran al 3%.

Després d'investigar, s'arriba a la conclusió que si s'ordenen els valors d'àrees per mides, i s'utilitza la moneda més gran per dividir a la resta, totes aquelles proporcions amb un valor superior a 1 no seran necessàries, perquè al dividir una àrea més gran per una més petita, serà impossible que surti un número per sobre d'1. Això permet treure molts valors que podrien causar errors.

Es fan moltes fotografies amb la Càmera Web, aquesta recolzada sobre un trípode, i es veu que comença a funcionar amb les monedes de 2€, 1€ i 50 ct.

S'observa que amb el fons blanc hi ha moltes ombres que creen soroll i que s'ha de controlar molt la llum, i per això es decideix canviar a un fons negre.



Figura 35. Fotografia a una moneda de dos euros i un euro a sobre un fons blanc. Es pot observar l'ombra. Font: Imatge de l'autor (24-8-20)



Figura 36. Fotografia a una moneda de dos euros i un euro a sobre un fons negre. Es casi impossible d'observar l'ombra. Font: Imatge de l'autor (24-8-20)

Es localitza una persona que té un grau mig de sistemes microinformàtics i xarxes i el grau superior de desenvolupador d'aplicacions multiplataforma perquè expliqui com compilar⁶ l'aplicació per a utilitzar-la al mòbil. S'instal·la correctament el Buildozer (veure annex 1), però no s'observa que la llibreria OpenCV no permet compilar. Es decideix posposar per a un futur, perquè sinó significaria canviar de llibreria, que voldria dir canviar tota l'aplicació.

S'utilitza Photoshop per a analitzar les imatges capturades, i s'observa que la càmera web crea distorsions a la imatge. Això explica perquè només funciona quan les monedes estan alineades sota la càmera. Per tal d'evitar aquestes distorsions, es decideix treballar amb càmera de fotos. S'observa que funciona i que detecta gairebé a la perfecció les monedes de 2€, 1€ i 50 ct a qualsevol punt de la càmera. Amb aquestes fotografies i el Photoshop, s'investiga quins són els paràmetres que millor funcionen, superposant la imatge original amb la processada i observant com varien l'una respecte l'altra. Com es pot observar a la figura 37, la mida de la línia blava equival al diàmetre de la moneda A que estava al centre. A la moneda B, que estava a l'extrem de la fotografia, se l'hi ha hagut d'afegir un troç de línia groga, pel que s'enten que, malgrat les dos siguin monedes de 1 euro i tinguin el mateix diàmetre, la càmera capta un diàmetre més gran en el cas de la moneda B.

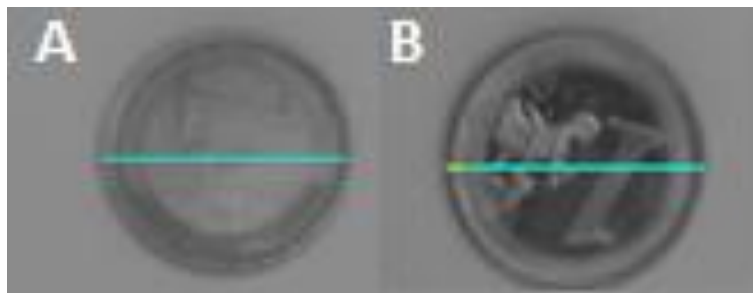


Figura 37. Fotografia feta amb la Càmera Web a una distància de 25 cm. Font: Imatge de l'autor (16-5-20)

Com a inconvenient, s'observa que la pantalla principal no s'actualitza després d'aplicar el processat d'imatges, pel que es decideix buscar una alternativa. Després d'investigar, es descobreix la funcionalitat *Screen*, que permet l'actualització de pantalla a diferència de la funcionalitat *Pop Up*, que és el que s'havia fet fins ara.

Un cop l'aplicació funciona correctament amb les monedes de 2€, 1€ i 50 ct, es decideix intentar aconseguir que funcioni amb totes les monedes. Per fer-ho, s'analitza la taula de proporcions mestres (veure figura 38). Observant la taula, es formula la hipòtesis de que si s'estableix manualment amb quina línia ha de treballar l'aplicació (basant-se amb la moneda més gran), s'eliminen per complet totes les proporcions inferiors al 4%. Es fan proves amb 4 monedes i s'observa que funciona.

⁶ Passar el codi escrit en llenguatge de programació a codi binari per tal de que l'ordinador pugui executar-lo.

		Moneda a detectar							
		2 EUR	50 ct	1 EUR	20 ct	5 ct	10 ct	2 ct	1 ct
Moneda mestra	2 EUR	1	0,887	0,815	0,747	0,681	0,588	0,53	0,398
	50 ct	1,128	1	0,919	0,842	0,768	0,663	0,598	0,449
	1 EUR	1,227	1,088	1	0,916	0,835	0,722	0,65	0,488
	20ct	1,339	1,188	1,092	1	0,912	0,788	0,71	0,533
	5 ct	1,468	1,302	1,197	1,096	1	0,864	0,779	0,585
	10 ct	1,7	1,508	1,386	1,269	1,158	1	0,901	0,677
	2 ct	1,886	1,673	1,538	1,408	1,284	1,11	1	0,751
	1 ct	2,511	2,227	2,047	1,875	1,71	1,477	1,331	1

Figura 38. Taula proporcions mestres. Font: Imatge de l'autor (29-05-20)

		Moneda a detectar							
		2 EUR	50 ct	1 EUR	20 ct	5 ct	10 ct	2 ct	1 ct
Moneda mestra	2 EUR	0%	11,3%	8,1%	8,3%	8,8%	13,7%	9,9%	24,9%
	50 ct		0%	8,10%	8,38%	8,79%	13,67%	9,80%	24,92%
	1 EUR			0%	8,40%	8,84%	13,53%	9,97%	24,92%
	20 ct				0%	8,80%	13,60%	9,90%	24,93%
	5 ct					0%	13,60%	9,84%	24,90%
	10 ct						0%	9,90%	24,86%
	2 ct							0%	24,9%
	1 ct								0%

Figura 39. Taula amb el % de diferència entre proporcions. Font: Imatge de l'autor (17-08-20)

Es cerca com implementar una funció que llegeixi els resultats en veu alta i s'intenta aplicar al sistema, però hi ha un error amb les llibreries i es deixa apartat per més endavant.

Es creen uns paràmetres estàndards per a l'aplicació, i es fan proves per a validar el seu funcionament.

Un cop finalitzada l'aplicació, es torna a intentar implementar el lector de veu del resultat. Es contacta amb l'expert en desenvolupament d'aplicacions una altra vegada per demanar consell. Finalment s'aconsegueix afegir a l'aplicació el lector de veu que llegirà en veu alta el resultat de la suma total.

Quan es considera que el projecte està acabat, es decideix buscar un nom per a l'aplicació. S'analitzen quines podrien ser bones opcions, i finalment s'escull COINT€R. S'escull aquest nom perquè és la combinació entre moneda (*coin*) i comptador (*counter*) amb anglès (per fer-lo més internacional). A més, es juga amb el signe € per donar a entendre que el que es compten són euros.

6.1.5 Descripció de l'aplicació

Per aconseguir un bon disseny es fa una recerca exhaustiva dels diferents editors de fotos populars en l'actualitat, de tal manera que es pugui combinar una aplicació útil, senzilla d'utilitzar i alhora estètica visualment. Després de buscar, es creu que la millor opció és un disseny semblant al de l'editor de fotos Lightroom, perquè permet una fàcil experiència a través de l'aplicació.

Per poder aconseguir que l'accés al contingut de l'aplicació sigui fàcil i pràctic, es creen unes finestres per navegació per tal de facilitar-ne l'ús (veure figura 44).

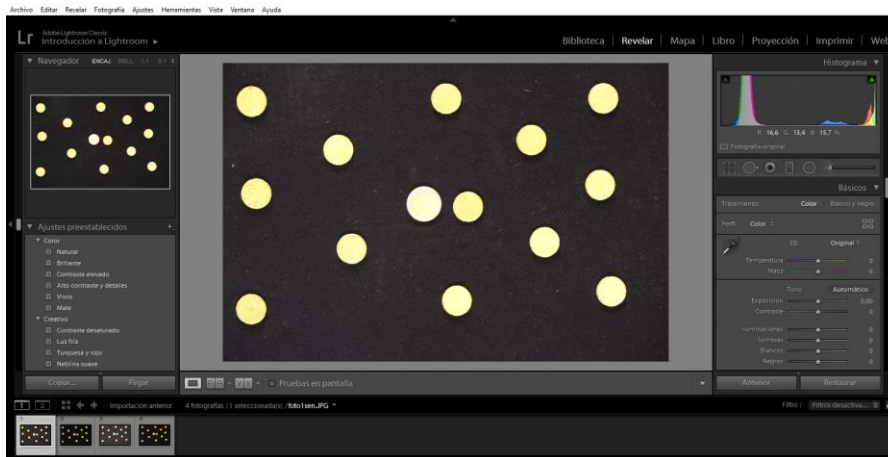


Figura 40. Editor de fotos Lightroom. Font: Imatge de l'autor (15-8-2020)

Per evitar errors d'entrada de dades del sistema, s'han establert mecanismes per controlar les variables que l'usuari pot entrar com serien les llistes desplegable (veure figura 41) els controls lliscants (veure figura 42) i les caselles de selecció (veure figura 43).



Figura 41. Les llistes desplegable només permeten triar els valors d'una llista predefinida. Font: Imatge de l'autor (15-8-2020)



Figura 42. Els controls lliscants permeten triar el valor en funció de la posició del botó sobre la barra de desplaçament. Els valors màxim i mínims queden predefinits. Font: Imatge de l'autor (15-8-2020)



Figura 43. Les caselles de selecció permeten escollir si es vol activar una funció o no. Font: Imatge de l'autor (15-8-2020)

Les operacions de processament (desenfocament, llindar,...) es van sobreescrivint sobre una imatge diferent a l'original; d'aquesta forma sempre es pot recuperar la imatge inicial. També s'han creat els botons de previsualització que permeten visualitzar l'efecte del processament però sense afectar al contingut de la imatge temporal. Si es prem el botó acceptar llavors es guarden els canvis sobre la imatge temporal, mentre que el botó cancel·lar surt de l'operació de processament sense guardar els canvis

El codi de programació s'adjunta a l'annex 2 i 3.

6.1.6 Pantalla principal

L'aplicació es compon d'una pantalla principal que dona accés a les diferents pantalles de processament i extracció de resultats. Aquesta s'ha dividit en tres àrees verticals:

- A l'esquerra, estan ubicades les eines de l'aplicació, obrir arxius, guardar la imatge processada en un nou arxiu, accés a captura d'imatges a través de càmera de vídeo.

-Al centre s'observa la imatge que s'està editant .

-A la dreta els botons d'accés a les funcions de processat d'imatge i extracció de dades: canals, rotació, desenfocament, llindar, erosió i dilatació, enfocament, costats i monedes.

Quan s'aplica una de les funcions de processat, s'actualitza la imatge de la pantalla principal.

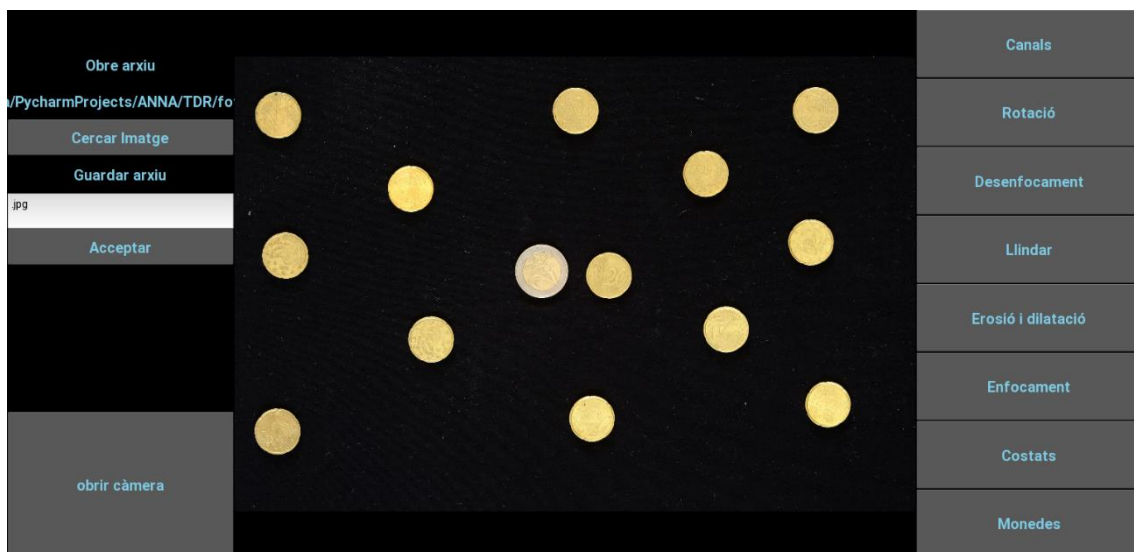


Figura 44. Pantalla principal de l'aplicació. Font: Imatge de l'autor (31-8-2020)

6.1.7 Captura Càmera Web

Aquesta pantalla permet capturar una imatge instantània sense haver de recórrer a una de l'arxiu, és a dir, a temps real. Per aconseguir-ho, s'ha fet servir la funció **Video Capture**. En aquesta aplicació, per poder capturar la imatge s'ha de prémer la tecla "q".

S'adjunta la funció de Python:

```
cv2.VideoCapture(0)
```

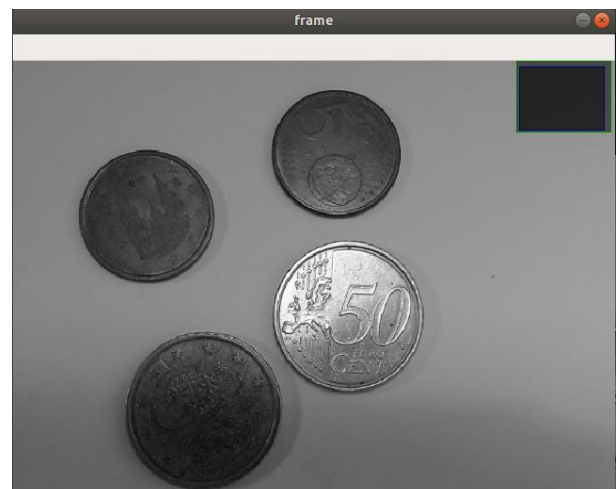


Figura 45. Pantalla de la funció càmera. Font: Imatge de l'autor (31-8-2020)

6.1.8 Canals

Aquesta pantalla permet poder seleccionar i manipular les diferents matrius que conformen l'espai de color seleccionat de treball. Així es poden visualitzar per separat cada una de les capes i triar la que té més informació per tal de realitzar el processat. En el cas del blanc i negre és una transformació a una escala de grisos.

6.1.8.1 RGB

En aquesta pantalla es pot seleccionar amb quin canal RGB es vol treballar. A més, permet incrementar o reduir el valor del color canal amb els controls lliscants.

Ha estat necessari programar la funció per a cada una de les possibles combinacions de canals de color seleccionats.

A mode d'exemple, s'adjunta la funció de Python quan es selecciona la capa vermella:

```
zeros = np.zeros(fotor.shape[:2], dtype="uint8")
```

Primer es crea una matriu de **zeros** amb les mateixes dimensions que la imatge per tal d'eliminar la informació del canal verd i blau.

```
B, G, R = cv2.split(fotor)
```

Després es separen els canals B, G, **R**, per tal d'extreure la informació del canal vermell, que correspon a **R**.

```
fotor = cv2.merge([zeros, zeros, R])
```

Finalment es recomposa la imatge final (**fotor**) amb només la informació del canal R.

Si llavors es vol incrementar el valor del canal vermell:

```
one= np.ones (fotor.shape[:2], dtype="uint8")
```

Primer es crea una matriu de uns (**one**) amb les mateixes dimensions que la imatge, que permet sumar un determinat valor (**vermell**) a la capa R. Abans, també s'ha de seguir el procediment **split** de separar la imatge en canals.

```
R = cv2.add(R, one * int(vermell))
```

A cada punt del canal R se li suma el valor seleccionat (**vermell**).

Posteriorment es recomposa la imatge amb el procés **merge** mencionat anteriorment.

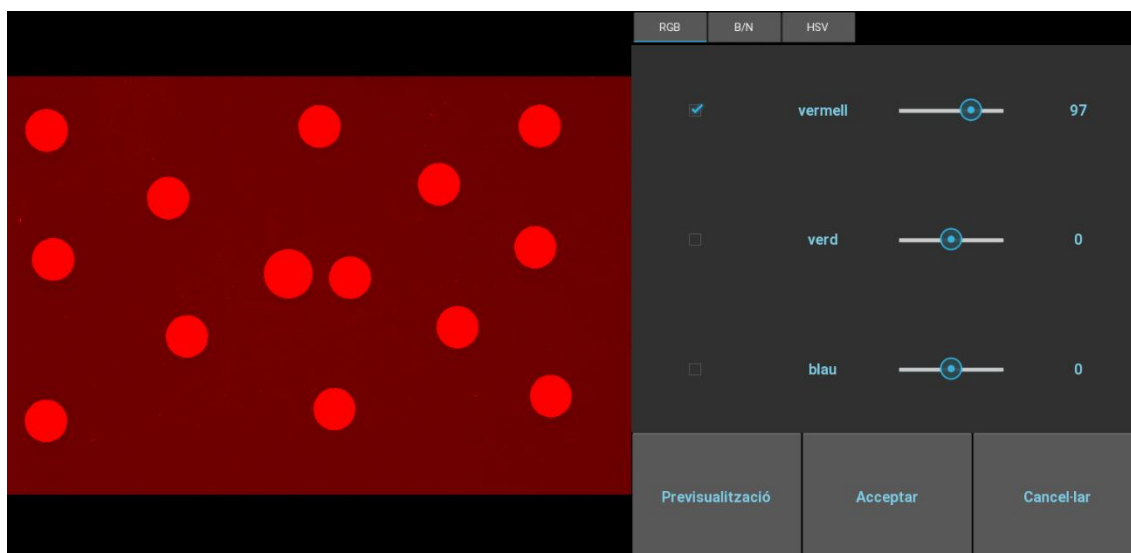


Figura 46. Pantalla corresponent a la finestra canals funció RGB. Font: Imatge de l'autor (31-8-2020)

6.1.8.2 B/N

La pestanya blanc i negre activa la conversió a escala de grisos. En aquest cas, no és necessari especificar cap paràmetre. Aquesta funció aplica un filtre de conversió anomenat **Color BGR2Gray**.

S'adjunta funció utilitzada en Python:

```
cv2.cvtColor(fotor, cv2.COLOR_BGR2GRAY)
```

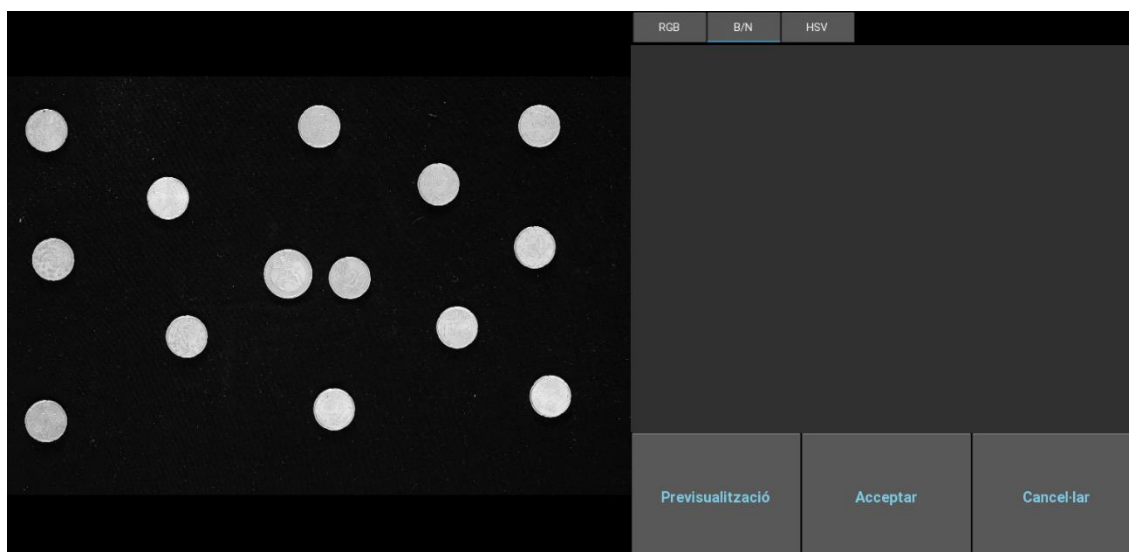


Figura 47. Pantalla corresponent a la finestra canals funció blanc i negre. Font: Imatge de l'autor (31-8-2020)

6.1.8.3 HSV

En aquesta pantalla es pot seleccionar amb quin canal HSV es vol treballar. S'aplica un filtre de conversió inicial anomenat **BGR2HSV**, que passa la imatge sobre la qual s'està treballant a HSV. Llavors segueix el mateix procés mencionat a la selecció de canals RGB (**split** i **merge**).

S'adjunta funció utilitzada en Python:

```
cv2.cvtColor(fotor, cv2.COLOR_BGR2HSV)
```

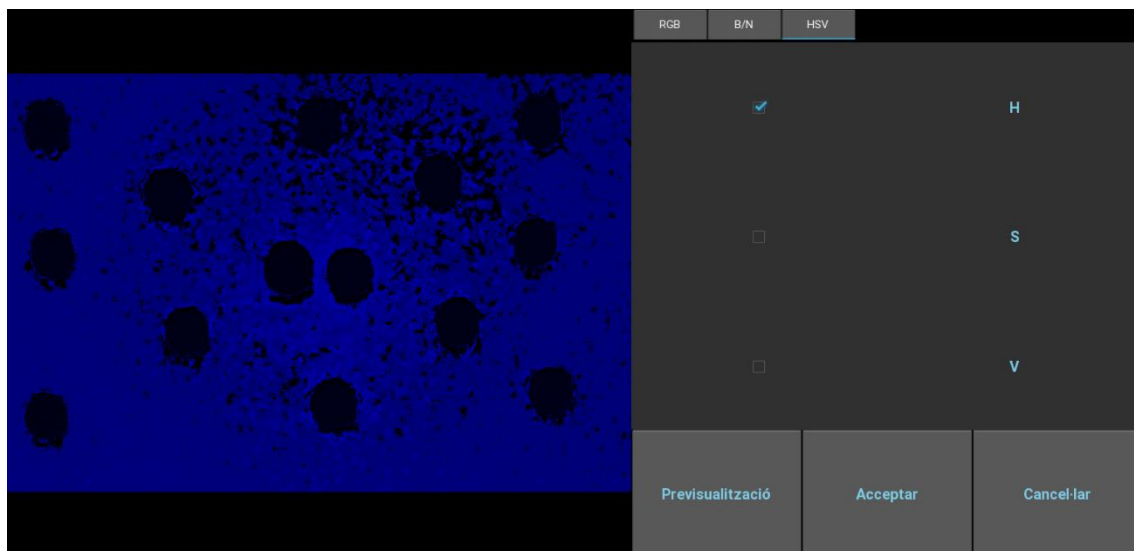


Figura 48. Pantalla corresponent a la finestra canals funció HSV. Font: Imatge de l'autor (31-8-2020)

6.1.9 Rotació

Aquesta pantalla permet canviar la rotació de la imatge, escalar-la i invertir-la verticalment o horitzontalment.

Escala: La imatge pot ser o massa petita o massa gran

Gir: L'orientació de la imatge pot no ser el correcte.

El sistema permet quin angle es vol rotar (**angle**) i si s'ha d'escalar (**escala**).

S'adjunta funció utilitzada en Python:

```
rotation_matrix = cv2.getRotationMatrix2D((int(amplada/2),
int(alcada/2)), int(angle), float(escala))
```

Pel que fa la funció invertir imatge, es pot escollir si es vol una transformació vertical (**0**) o horitzontal (**1**).

S'adjunta funció utilitzada en Python:

```
cv2.flip(rotated_image, 0)
```

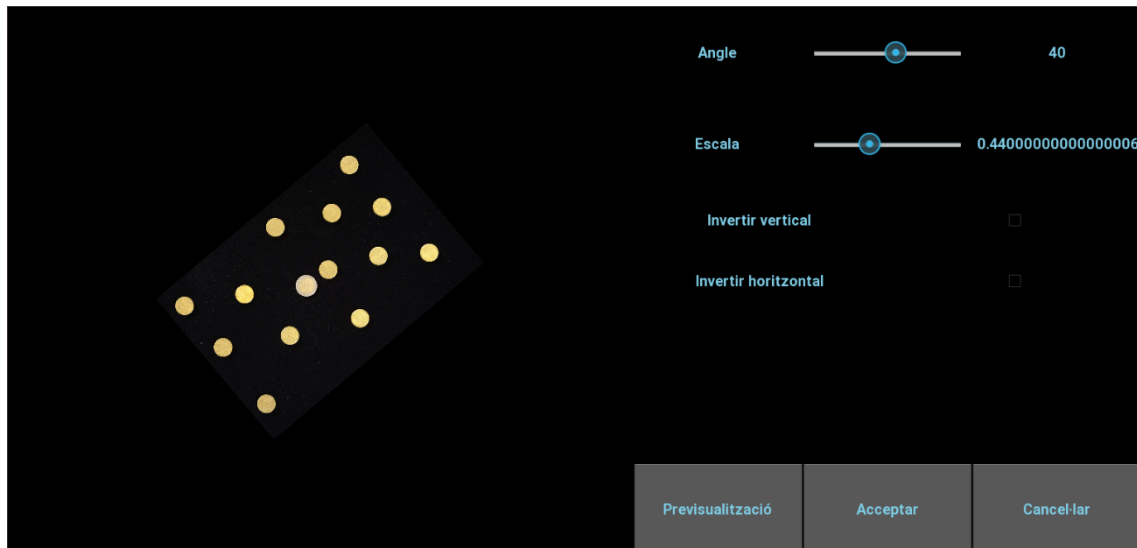


Figura 49. Pantalla corresponent a la finestra rotació. Font: Imatge de l'autor (31-8-2020)

6.1.10 Desenfocament

Aquesta pantalla de l'aplicació permet treballar amb el desenfocament aplicat a una imatge. En l'aplicació desenvolupada és important perquè el càlcul d'àrees es basa en la determinació dels costats, per tant és molt important que aquests destaquin, i la resta de punts es facin més borrosos. En aquest treball, s'han utilitzat tres tipus de desenfocament diferents: Median, Gaussian i Bilateral.

6.1.10.1 Median

El sistema permet seleccionar el rang (**median**) de la matriu de transformació. Com més gran sigui aquesta, més gran serà l'efecte del desenfocament i menys nítida serà la imatge, perquè hi hauran més punts que es tindran en compte per fer la mitjana, per tant, serà menys precís.

S'adjunta funció utilitzada en Python:

```
cv2.blur(fotor, (int(median), int(median)))
```

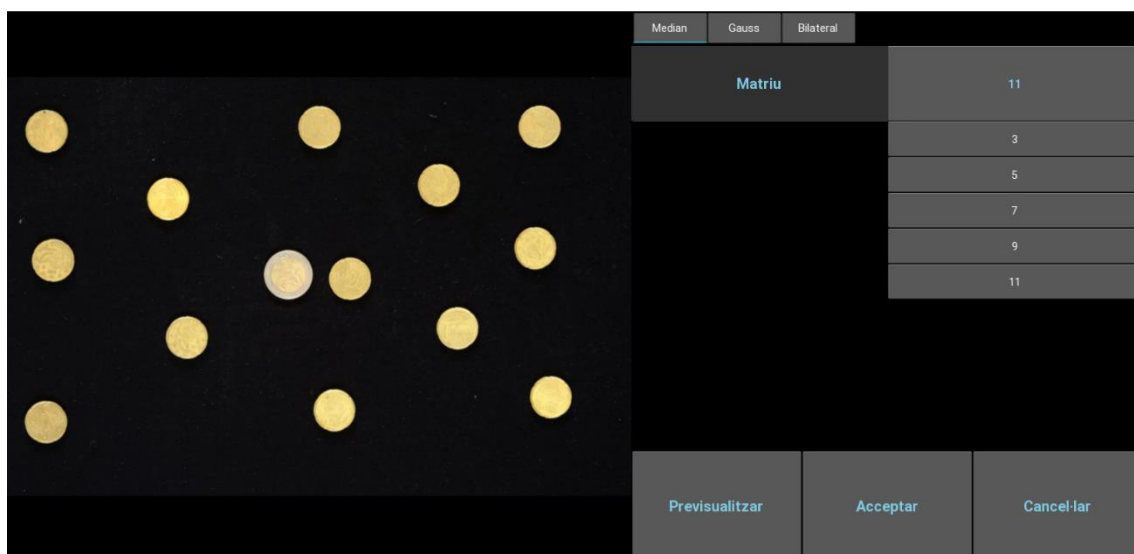


Figura 50. Pantalla corresponent a la finestra desenfocament funció Median. Font: Imatge de l'autor (31-8-2020)

6.1.10.2 Gaussian

El sistema permet escollir el rang (**gauss**) de la matriu de transformació. Com més gran sigui aquesta, més gran serà l'efecte del desenfocament i menys nítida serà la imatge, perquè hi hauran més punts que es tindran en compte per fer la mitjana, per tant, serà menys precís.

S'adjunta funció utilitzada en Python:

```
cv2.GaussianBlur(fotor, (int(gauss), int(gauss)), 0)
```

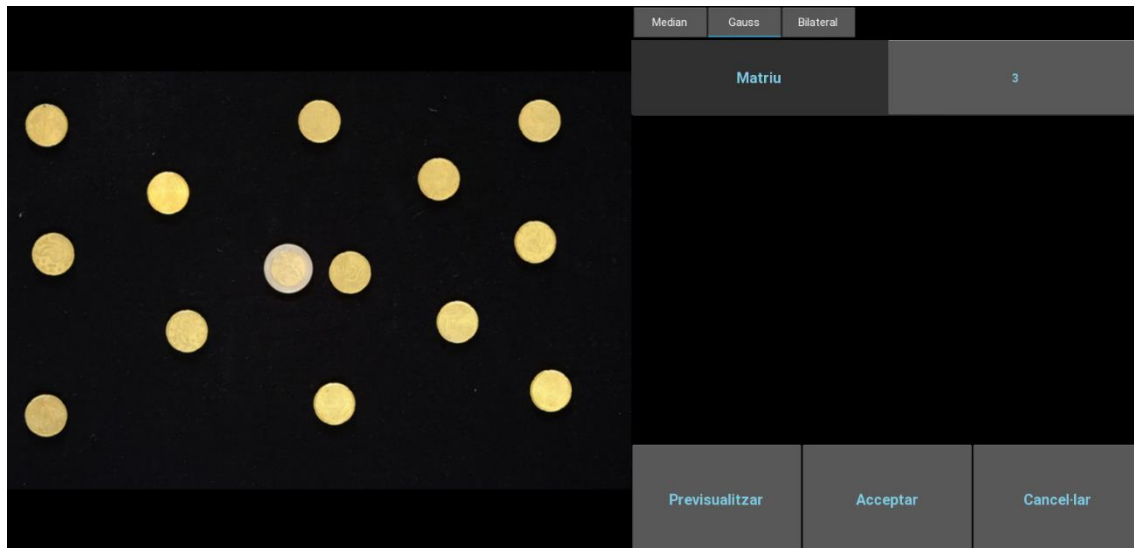


Figura 51. Pantalla corresponent a la finestra desenfocament funció Gaussian. Font: Imatge de l'autor (31-8-2020)

6.1.10.3 Bilateral

El sistema permet escollir:

- El valor del radi (**radi**), que seria el diàmetre d'actuació de la matriu de transformació.
- El valor del color del sigma (**sigmac**), que serveix per a determinar a partir de quina diferència de colors barreja o no els píxels.
- El valor de sigma x (**sigma**), que serveix per especificar si es vol conservar poc o molt el contorn.

S'adjunta funció utilitzada en Python:

```
cv2.bilateralFilter(fotor, int(radi), int(sigmac), int(sigma))
```

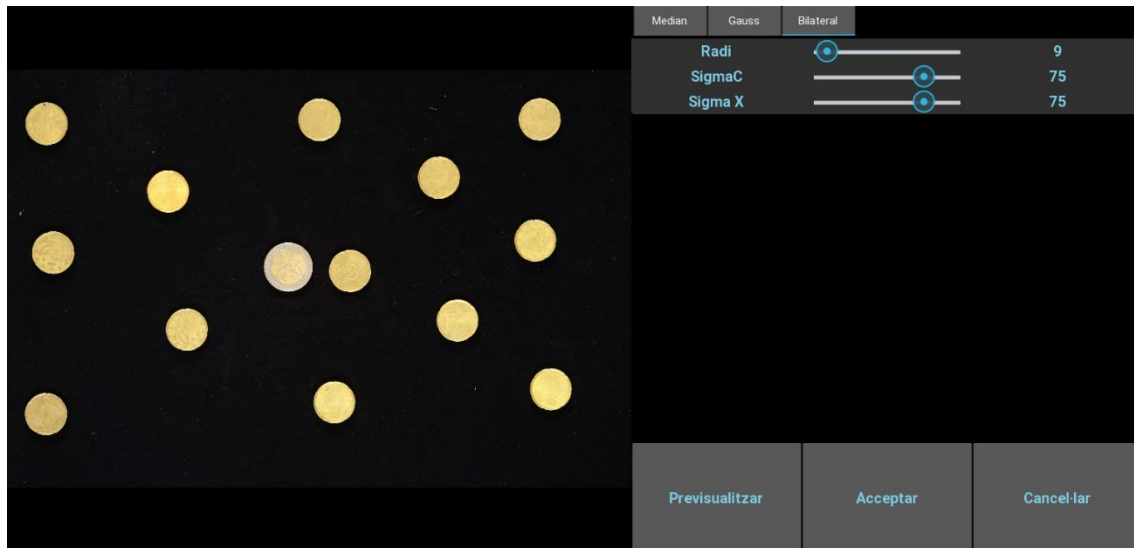



Figura 52. Pantalla corresponent a la finestra desenfocament funció Bilateral. Font: Imatge de l'autor (31-8-2020)

6.1.11 Llindar

Aquesta pantalla de l'aplicació permet un processat d'imatge d'acord amb un valor de llindar. És necessari perquè el llindar transforma els píxels en blanc o negre, i d'aquesta forma facilita la detecció de contorns.

El control lliscant permet triar el llindar mínim de forma que tots els valors superiors o iguals els transforma a blanc mentre que els inferiors a negre.

S'aplica el filtre **thresh binary** a la imatge (**fotor**), tenint en compte el valor (**umbral**) establert.

S'adjunta funció utilitzada en Python:

```
ret, thresh1 = cv2.threshold(fotor, int(umbral), 255, cv2.THRESH_BINARY)
```

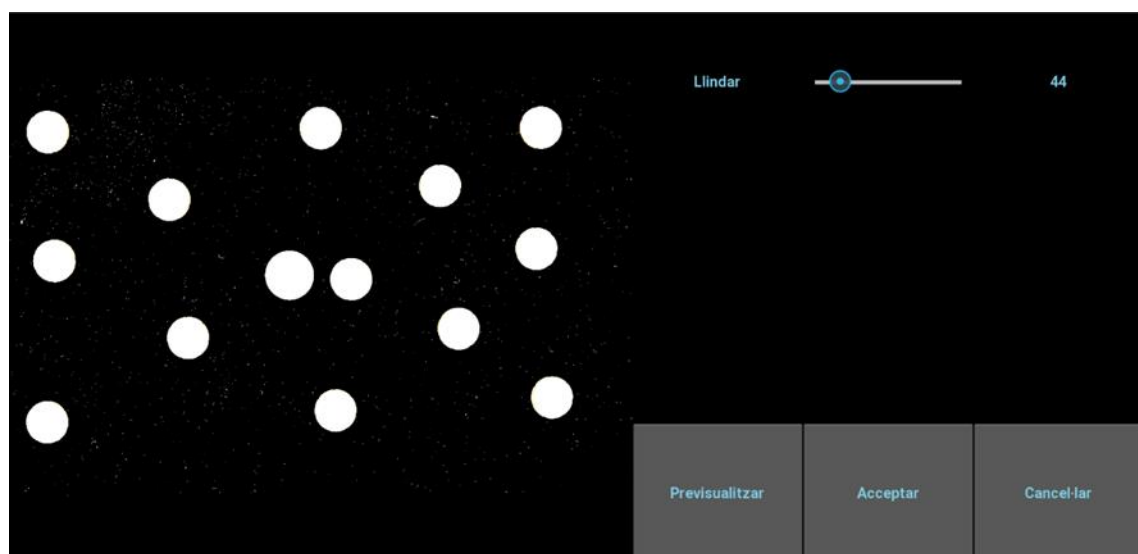


Figura 53. Pantalla corresponent a la finestra llindar. Font: Imatge de l'autor (31-8-2020)

6.1.12 Erosió i dilatació

Aquesta pantalla treballa amb les eines de erosió i dilatació. S'ha inclòs en el treball perquè aquest es basa en la detecció de costats, per la qual cosa, qualsevol soroll de la imatge podria causar una detecció errònia. A més, permet minimitzar l'impacte de les ombres.

L'eina erosió: Serveix per aconseguir eliminar punts en la fotografia que no són costats de monedes.

L'eina dilatació: Serveix per augmentar la mida dels punts que hi ha a la imatge. Això és útil en el treball degut a que en la fotografia pot haver-hi una manca o excés de llum que faci que un costat d'una moneda no es detecti bé o sigui molt petit. Amb l'eina dilatació s'augmenta la mida dels punts del costat per evitar que això passi.

El sistema permet triar la mida de la matriu (**kernel**) de la transformació, quina transformació es vol aplicar (**erosion**) i quantes vegades (**sv1**).

En el cas de la dilatació s'utilitza la funció **dilate** en comptes de **erode**.

S'adjunta funció utilitzada en Python:

```
cv2.erode(erosion, kernel, iterations=int(sv1))
```

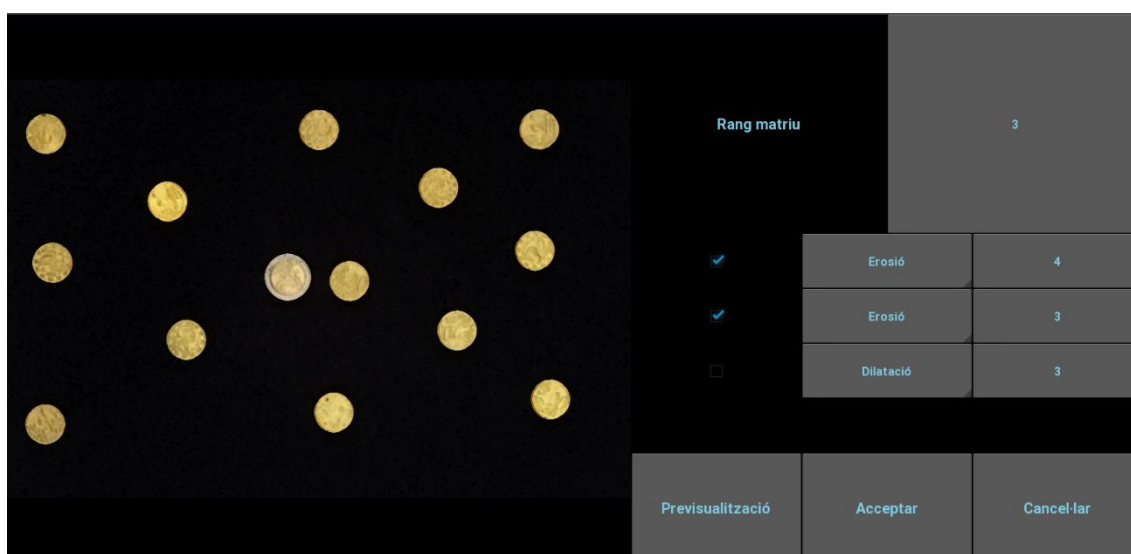


Figura 54. Pantalla corresponent a la finestra erosió i dilatació. Font: Imatge de l'autor (31-8-2020)

6.1.13 Enfocament

Aquesta pantalla disposa de tècniques d'enfocament. Si la imatge capturada té molt mala resolució, aquesta funció pot millorar-la, ja que augmenta el contrast entre els píxels, permetent que els diferents objectes puguin diferenciar-se millor. En aquest cas el sistema permet treballar amb dos tipus de filtres: Enfocament i Pas alt.

S'aplica la funció **filter 2D**, que permet aplicar a la imatge (**fotor**) la matriu (**kernel sharpening**) desitjada.

S'adjunta funció utilitzada en Python:

```
cv2.filter2D(fotor, -1, kernel_sharpening)
```

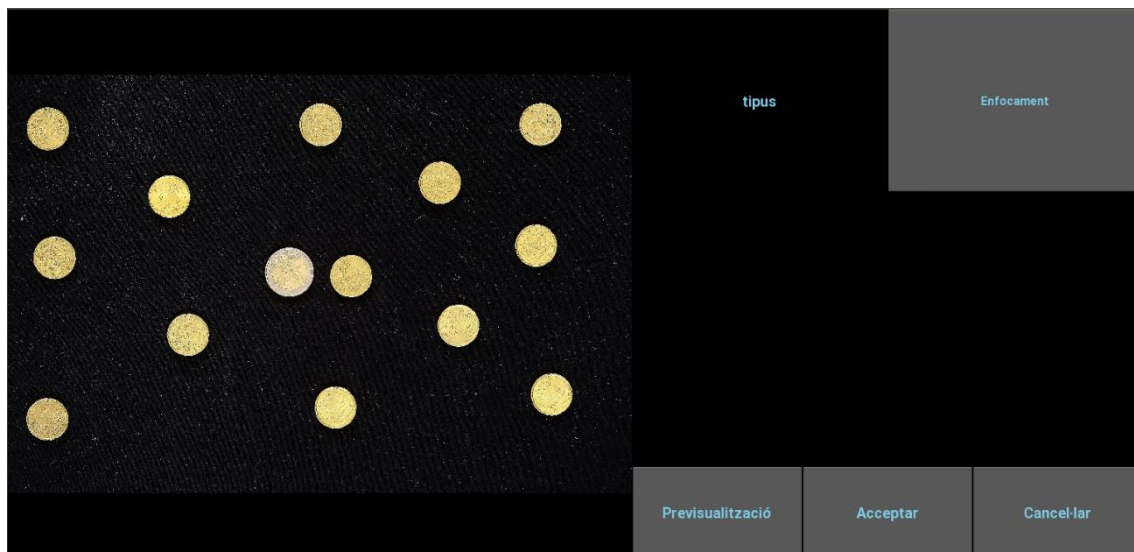


Figura 55. Pantalla corresponent a la finestra enfocament. Font: Imatge de l'autor (31-8-2020)

6.1.14 Costats

Per poder detectar els costats de les monedes, s'ha hagut d'aplicar la funció detecció de costats. Per detectar-los es poden fer servir tres eines diferents:

6.1.14.1 Canny

El sistema permet establir amb una barra el gradient mínim (**gradd**) i la detecció de gradient (**gradm**).

S'adjunta funció utilitzada en Python:

```
cv2.Canny(fotor, int(gradd), int(gradm))
```

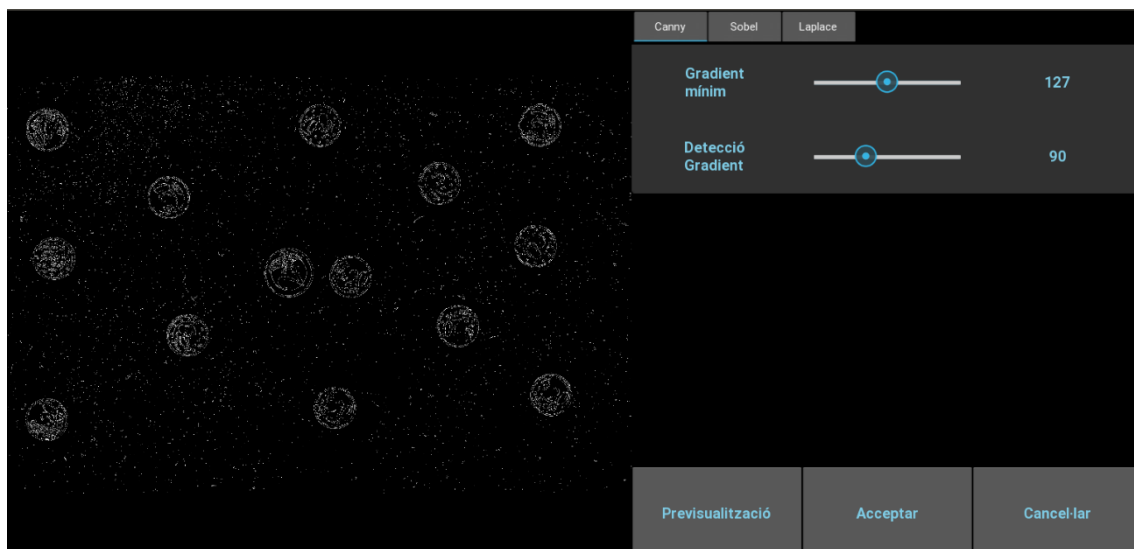


Figura 56. Pantalla corresponent a la finestra costats funció Canny. Font: Imatge de l'autor (31-8-2020)

6.1.14.2 Sobel

El sistema permet escollir el rang (**sobel_m**) de la matriu de transformació que s'aplica que s'aplica i si es vol utilitzar en direcció x (0,1) i/o Y (1,0).

S'adjunta funció utilitzada en Python:

```
cv2.Sobel(fotor, cv2.CV_64F, 1, 0, ksize=int(sobelm))
```

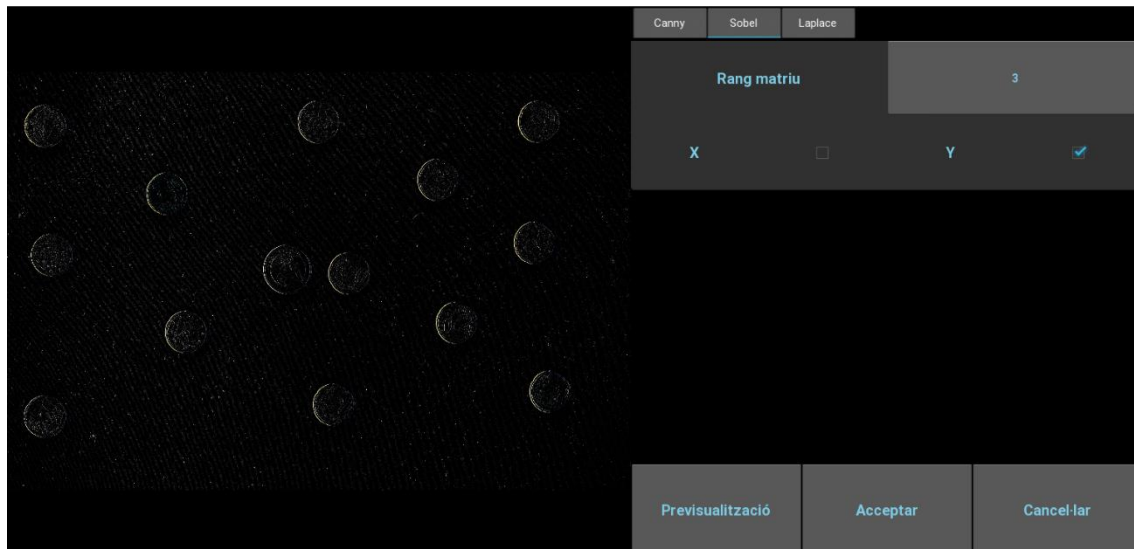


Figura 57. Pantalla corresponent a la finestra costats funció Sobel. Font: Imatge de l'autor (31-8-2020)

6.1.14.3 Laplace

El sistema permet escollir el rang (**laplaces**) de matriu de transformació.

S'adjunta funció utilitzada en Python:

```
cv2.Laplacian(fotor, cv2.CV_64F, ksize=int(laplaces))
```

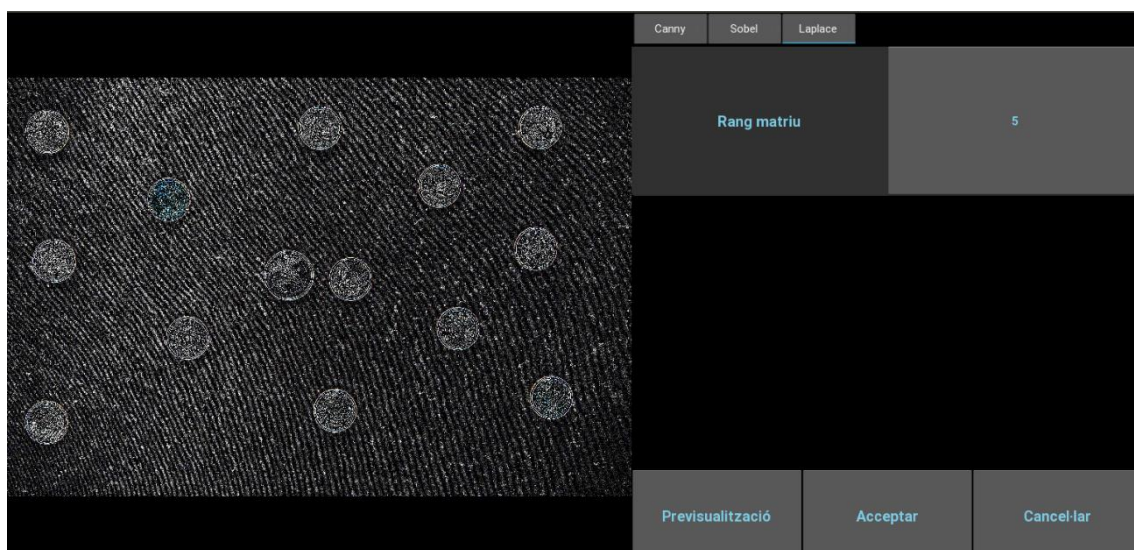


Figura 58. Pantalla corresponent a la finestra costats funció Laplace. Font: Imatge de l'autor (31-8-2020)

6.1.15 Monedes

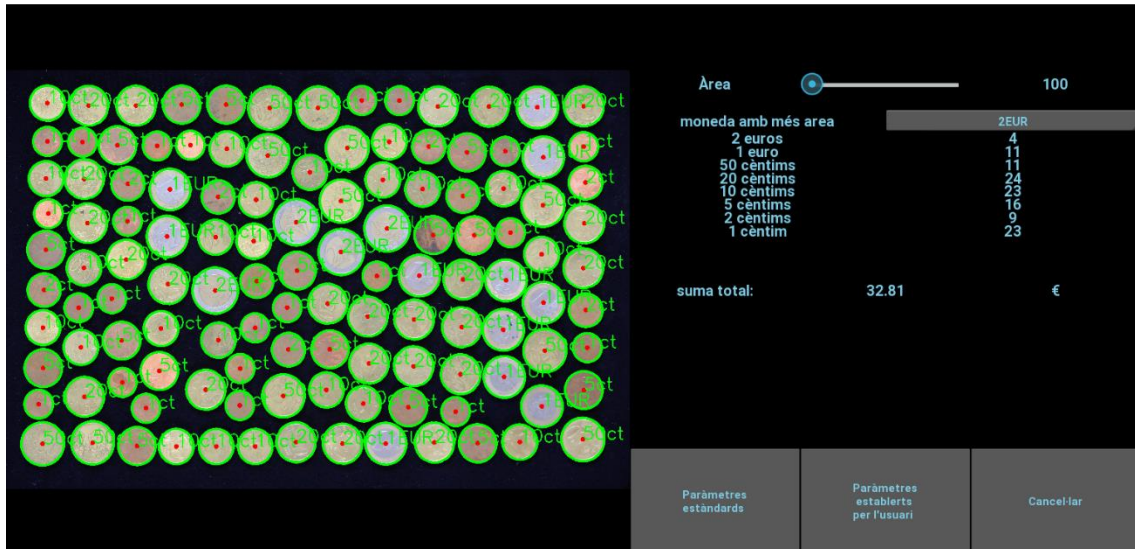


Figura 59. Pantalla corresponent a la finestra monedes. Font: Imatge de l'autor (31-8-2020)

Aquesta pantalla és la d'extracció de resultats i l'aplicació de comptar monedes com a tal.

Es defineix quina és la moneda de més mida manualment i a partir de quina àrea no es considera soroll.

Per obtenir el càlcul de monedes es poden utilitzar dos botons: el botó paràmetres establerts per l'usuari en el qual es processa la imatge que l'usuari ha modificat prèviament, i el botó paràmetres estàndards en el qual s'apliquen uns paràmetres predefinits de processament.

El programa consisteix en dividir les àrees detectades a la imatge i comparar-les amb les dades de la taula de proporcions mestres (veure figura 38). Es permet un error de fins el 4% ja que com s'ha mencionat a l'apartat 7.1.4, les àrees poden variar segons grau d'enfocament, ombres, quantitat de llum, error de la càmera.

Un cop s'ha trobat la suma total de monedes, un lector de veu llegeix el resultat (**suma total**) en veu alta utilitzant la funció *speak*.

S'adjunta funció utilitzada en Python:

```
speak("tens" + str(sumatotal)+"euros")
```

Donada la complexitat del codi, s'ha considerat ajuntar un diagrama de flux per facilitar la seva comprensió.

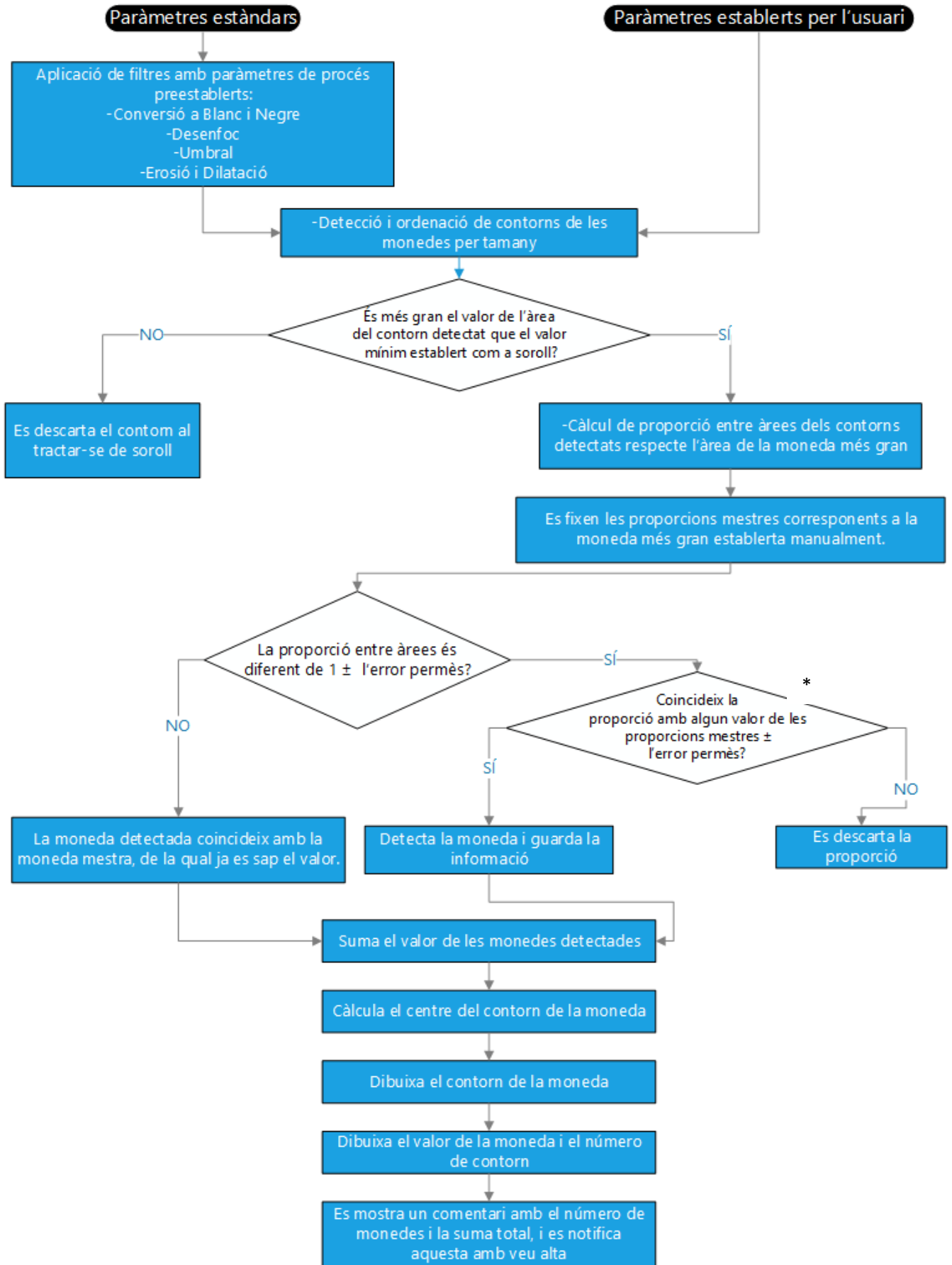


Figura 60. Diagrama de flux on s'explica el codi monedes. Font: Imatge de l'autor (5-9-20)

*


```
*
for (i, c) in enumerate(range(len(ratiose))):
    #for l in range(len(master)):
    for z in range(len(master[0])):
        if (ratiose[c] >= (master[l][z]) * (1 - tol) and ratiose[c] <= (master[l][z]) * (1 + tol) and (
            ratiose[c] < (1 - tol) or ratiose[c] > (1 + tol)):
            idcountor[c + 1] = c + 1
            idarea[c + 1] = monedes[z]
            #print("Moneda referencia", monedes[l], "moneda a determinar", monedes[z])
            zz.append(monedes[z])
            ratiotrobat = 1
            ratiosef.append(ratiose[c])
            errorr.append(abs((master[l][z]- ratiose[c])/master[l][z]*100))
```

Figura 61. Part del codi corresponent a l'explicació feta en el requadre amb l'asterisc del diagrama de flux. Font: Imatge de l'autor (5-9-20)

La figura 61 correspon a una petita part la funció monedes. En aquest *for*⁷ és on s'identifica la moneda. Primer s'agafen tots els valors de la llista **ratiose** (que serien les proporcions creades). Tot seguit es comparen amb tots els valors de la línia de proporcions mestres definides manualment (**master**).

En el cas que trobi una proporció que coincideixi amb el valor de **master**, tenint en compte la tolerància establerta (**tol**), s'afegeix a la posició de la llista **idcountor** c+1 (on c és equivalent a la posició del valor de **ratiose** analitzat) per tal determinar quin contorn l'ha generat i a la llista **idarea**, a la posició c+1, el valor de la moneda detectada (**monedes**).

També es determina quina és la moneda de referència en base a la línia escollida de **master**. Finalment, s'afegeix la moneda trobada en una llista anomenada **zz**.

6.2 Verificació del funcionament de l'aplicació

La visió artificial es basa en el procediment de processat i obtenció de resultat. S'ha de tenir present que el procés de captació de la imatge condiciona la informació de la imatge, de fet, es podria dir que és pràcticament impossible fer 2 fotografies iguals, ja que la informació pot variar depenent de la posició del focus de llum, l'angle, el punt d'enfocament de la lent, el temps d'exposició, la sensibilitat de la pel·lícula,... Aquestes proves intenten demostrar que malgrat la informació captada pel sensor és diferent en cada fotografia, aplicant diferents tècniques de processat l'aplicació és capaç de funcionar correctament. S'utilitzaran els paràmetres estàndards que s'han anat fent servir durant el desenvolupament de l'aplicació, però en el cas de que no funcionin, s'intentarà trobar uns de nous que permetin fer-ho.

S'han dissenyat 3 escenaris que es creuen que poden influir més sobre les dades extrems de la imatge: quantitat de llum, fons i diferents lents.

⁷ Un *for*, en programació, s'utilitza quan es vol repetir un conjunt d'instruccions un número de vegades finit.

Per poder realitzar les proves, s'ha utilitzat un trípode, una màrfega, una tela negra i dos tipus de càmeres (càmera mòbil Huawei p30 Lite, càmera Fuji Film XT 20 amb l'òptica 55-200).

Fons:

El fons pot crear interferències a l'hora de detectar monedes, per la qual cosa, s'ha volgut analitzar si funcionava amb diferents tipus de fons. S'ha fet la prova amb 3 tipus diferents (negre, blanc i amb relleu), amb diferents colors de monedes (2 EUR, 1 EUR, 50 ct i 5 ct) i amb diferents valors d'exposició o EV (0 i -1).

Quantitat de llum:

S'ha comprovat aquesta variable perquè la llum afecta als colors i a la mida de les ombres. S'han fet fotos a diferents valors d'exposició o EV (0, +1, -1, -2, -3) i amb diferents colors de monedes (2 EUR, 1 EUR, 50 ct i 5 ct).



Figura 62. Preparació de l'experiment.
Font: Imatge de l'autor (26-8-2020)

Diferents lents:

Depenent de la lent de la càmera es poden crear deformacions a l'hora de fer fotografies. S'ha considerat interessant analitzar quina era la zona que tenia menys error per tal d'utilitzar-la amb cada tipus de lent.

S'ha realitzat la prova amb 2 tipus de lents diferents: una càmera Fujifilm Xt-20 amb una lent 55-200 i amb un mòbil Huawei p30 Lite. Per la lent de la càmera Fujifilm s'ha utilitzat una moneda de 2 EUR i 13 monedes de 20 ct, i per la de la càmera del Huawei 1 moneda de 2 EUR i 13 monedes de 10 ct.

6.2.1 Resultats

De les proves realitzades prèviament s'han extret les següents conclusions:

Pel que fa al fons:

- L'aplicació permet comptar les monedes amb diferents fons.
- Només funcionen els paràmetres estàndards en el cas del fons negre.
- Si es canvia el fons s'ha de tenir en compte que s'hauran de realitzar uns ajustaments previs.
- Com més brillant és una moneda, més gran ha de ser el valor de el llindar per poder detectar el contorn d'aquesta.
- Es veu que com més clar sigui el fons, major valor haurà de tenir el llindar per poder detectar el contorn de la moneda.

Pel que fa a l'exposició:

- S'observa que permet detectar les monedes a diferents exposicions.
- Només funcionen els paràmetres estàndards en el cas d'EV 0, EV -1 i EV -2.

Pel que fa al tipus de lent:

- S'ha vist que la lent que millor funciona i menys distorsió crea és la de la càmera Fuji Film quan la distància focal és superior a 60mm. Un cop s'ha arribat a aquesta conclusió s'han afegit més de 120 monedes i les ha detectat totes correctament.
- Les fotografies fetes amb la lent de 60mm es poden processar correctament amb els paràmetres estàndards.
- En el cas del mòbil, que té un gran angular, la càmera crea menys distorsió a la zona central, i es detecten millor les monedes. En les zones laterals augmenta la distorsió i no sempre els resultats són correctes. En aquest cas, sempre s'hauran de configurar manualment les tècniques de processat abans de aconseguir una correcta detecció.
- Els paràmetres estàndards funcionen únicament en les fotografies fetes amb el mòbil amb EV -2, en la resta s'han de modificar manualment els paràmetres.

6.2.2 Accions de millora

Després de realitzar una avaluació de l'aplicació amb les característiques actuals, es planteja com a proposta de millora:

- Ampliar el nombre de variables per tal detectar automàticament la moneda més gran i minimitzar els errors. Aquestes poden ser: color, número de contorns a l'interior, forma...
- Ampliar els coneixements de programació per tal de poder compilar l'aplicació i executar-la des d'un mòbil, ja que en el model actual, només és possible des de l'ordinador.

6.3 Usos

L'objectiu inicial de la creació d'aquesta aplicació era el d'arribar a ser una eina útil per aquelles persones que per alguna raó no són capaces de comptar monedes per si soles, ja sigui perquè no poden o perquè els costa (nens petits, persones grans,...). Després d'haver explicat la idea a companys de l'autor del treball, s'ha observat que a molts d'aquests els interessaria l'aplicació per tal d'estalviar temps comptant monedes manualment; per tant, l'aplicació mateixa podria ser útil per a tothom.

I donat que, una part majoritària de l'aplicació es basa amb tècniques de retoc de la imatge, aquesta podria utilitzar-se com a editor fotogràfic.

7 Conclusions

L'aplicació funciona correctament i li és possible identificar monedes per la proporció entre les àrees de cadascuna amb unes condicions estàndards. Per aquesta raó, es creu assolit l'objectiu principal del treball de recerca. A més, s'ha incorporat un lector de veu que permetrà a persones amb dificultats per llegir, poder-la utilitzar.

Crear una aplicació amb Python capaç de reconèixer i sumar monedes de forma automàtica semblava totalment impossible fa uns set mesos, però gràcies als recursos d'Internet, ho he aconseguit. Això m'ha demostrat que aquest és una font d'informació realment molt potent si es sap utilitzar correctament, ja que he estat capaç d'aprendre un llenguatge de programació des de zero, i crear una aplicació només amb l'ajuda de vídeo tutorials de Youtube i la plataforma Udemy, entre d'altres.

A més, aquest treball m'ha permès entrar en un món com és el de la programació, pràcticament desconegut per mi, però de vigorosa actualitat. He vist que dedicar-me exclusivament a la programació no és una idea que m'atragui especialment, al viure en carn pròpia la dificultat i paciència que he vist que s'ha de tenir per poder dur a terme aquesta feina. Malgrat això, no descarto en un futur dedicar-me a un àmbit professional on la programació pugui ser una eina per aconseguir un fi.

M'he adonat de la importància de les tècniques de processat d'imatge, ja que aquestes poden marcar la diferència entre una detecció correcta o una d'incorrecta. Durant les proves de verificació de l'aplicació, he vist que per aquest treball n'hi ha algunes que són prescindibles, com seria el cas de l'enfocament, l'espai de color HSV i RGB i la rotació i escala, ja que aquests paràmetres no són dels més essencials.

Tanmateix, no només he descobert la importància de tècniques de processament d'imatge que desconeixia, sinó que també he pogut apreciar la importància d'una bona fotografia, i la dificultat que hi ha per aconseguir-la, ja que hi ha una quantitat de paràmetres, que poden afectar la informació que captarà el sensor. A més, he après el funcionament del sensor de la càmera, un procés que desconeixia.

En aquest treball de recerca he experimentat en viu l'anomenat mètode científic de l'assaig error. Potser l'hauria pogut evitar si des d'un inici hagués analitzat totes les possibilitats i hagués estudiat correctament tots els passos, en comptes d'haver de rectificar un cop ja s'ha començat la feina; per exemple, si jo des d'un inici hagués calculat les proporcions entre monedes, i hagués vist que hi havia una diferència més petita del 0,01% entre alguna d'elles, segurament hauria canviat totalment l'enfocament del projecte. No obstant, al equivocar-me i trobar-me en aquella situació, he hagut de buscar de una alternativa per a solucionar-ho, en comptes d'abandonar el projecte i fer-ne un altre.

He hagut de fer una recerca amb els meus recursos per poder solucionar cada problema que em trobava programant, passant-me hores i hores només per aconseguir un petit avenç. Al ser un procés tant llarg i dificultós, cada cop que aconseguia solucionar un problema, ho sentia com una gran victòria. Al haver dedicat

tant de temps i esforç, quan finalment l'aplicació ha funcionat, he sentit molta satisfacció, perquè tot el treball fet ha estat justament recompensat.

En síntesi, amb aquest treball he descobert que realment és possible programar una aplicació capaç de comptar monedes amb visió artificial, i també que encara es pot aplicar la visió artificial en molts més àmbits del dia a dia en què no s'utilitza. Per exemple, només retocant petites coses del codi d'aquesta aplicació, se'n poden crear moltíssimes més: comptadors de pastilles, comptadors de colònies de bacteris,...

8 Bibliografia

1. *Historia de la inteligencia artificial*. Diversos autors. Modificat 2020. Wikipedia. 6 de maig de 2020. https://es.wikipedia.org/wiki/Historia_de_la_inteligencia_artificial,
2. *Intel·ligència artificial*. Diversos autors. Modificat 2020. Viquipèdia. 15 de maig de 2020 https://ca.wikipedia.org/wiki/Intel%C2%B7lig%C3%A8ncia_artificial
3. *Quarta Revolució Industrial*. Diversos autors. Modificat 2019. Viquipèdia. 19 de maig de 2020. https://ca.wikipedia.org/wiki/Quarta_Revoluci%C3%B3_Industrial
4. *Màquina Enigma*. Diversos autors. Modificat 2019. Viquipèdia. 20 de maig de 2020. https://ca.wikipedia.org/wiki/M%C3%A0quina_Enigma
5. *Las medidas de seguridad*. Autor desconegut. Modificat 2019. Banco de España. 5 de maig de 2020. https://www.bde.es/bde/es/areas/billemon/Publico_general/Monedas_de_euro/seguridad/Las_medidas_de_seguridad.html
6. Eloi Maduell i Garcia. (2017). *Visió artificial [arxiu PDF]*. <https://www.playmodes.com/wp-content/uploads/2017/04/UOC-PDF-6.pdf>
7. *7 aplicaciones de la visión artificial*. Diversos autors. Modificat 2019 . Atria Innovation. 22 de maig de 2020. <https://www.atriainnovation.com/siete-aplicaciones-vision-artificial/>
8. *6 usos emergentes de la Visión Artificial en Retail*. David Cabañeros Blanco. Modificat 2020. Izertis. 22 de maig de 2020. <https://www.izertis.com/es/-/blog/6-usos-emergentes-de-la-vision-artificial-en-retail>
9. *Digital camera sensors*. Autor desconegut. Modificat 2020. Cambridge in colour. 10 de juny de 2020. <https://www.cambridgeincolour.com/tutorials/camera-sensors.htm>
10. Hedgecoe, John. *El nuevo libro de la fotografía: cómo ver y tomar mejores fotografías*. 1a edició. Barcelona: Blume, 1995.
11. *Fotón*. Diversos autors. Modificat 2020. Wikipedia. 10 de juny de 2020. <https://es.wikipedia.org/wiki/Fot%C3%B3n>
12. *Llum*. Diversos autors. Modificat 2020. Viquipèdia. 11 de juny de 2020. <https://ca.wikipedia.org/wiki/Llum>
13. *Llum*. Autor desconegut. Modificat 2019 Enciclopèdia.cat. 11 de juny de 2020. <https://www.enciclopedia.cat/ec-gec-0120427.xm>
14. *Longitud d'ona*. Diversos autors. Modificat 2020. Viquipèdia. 13 de juny de 2020. https://ca.wikipedia.org/wiki/Longitud_d%27ona
15. *Imatge digital*. Diversos autors. Modificat 2019. Viquipèdia. 13 de juny de 2020. https://ca.wikipedia.org/wiki/Imatge_digital
16. *Retina*. Diversos autors. Modificat 2020. Viquipèdia. 14 de juny de 2020. <https://ca.wikipedia.org/wiki/Retina>

17. *Processament d'imatges*. Diversos autors. Modificat 2015 . Viquipèdia. 16 de juny de 2020.
https://ca.wikipedia.org/wiki/Processament_d%27imatges
18. *Transformaciones morfológicas*. Autor desconegut. Modificat 2018. Unipython. 17 de juny de 2020.
<https://unipython.com/transformaciones-morfologicas/>
19. *Model de color HSV*. Diversos autors. Modificat 2018. Viquipèdia. 20 de juny de 2020.
https://ca.wikipedia.org/wiki/Model_de_color_HSV
20. *Soroll*. Diversos autors. Modificat 2020. Viquipèdia. 21 de juny de 2020.
<https://ca.wikipedia.org/wiki/Soroll>
21. *OpenCV Operaciones Morfológicas*. Autor desconegut. Modificat 2017. Tutor de programación. 23 de juny de 2020. <http://acodigo.blogspot.com/2017/04/opencv-operaciones-morfologicas.html>
22. *Umbralización de una imagen*. Autor desconegut. Modificat 2018. Unipython. 26 de juny de 2020.
<https://unipython.com/umbralizacion-una-imagen/>
23. *Laplace Operator*. Autor desconegut. Modificat 2019. OpenCV. 26 de juny de 2020.
https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/laplace_operator/laplace_operator.html
24. Computerphile. (2015, novembre 11). *Canny Edge Detector – Computerphile*. [Vídeo].
<https://www.youtube.com/watch?v=sRFM5IEqR2w&t=158s>
25. Computerphile. (2015, novembre 4). *Finding the Edges (Sobel Operator) – Computerphile*. [Vídeo].
26. *OpenCV*. Diversos autors. Modificat 2020. Wikipedia. 29 de juny de 2020.
<https://es.wikipedia.org/wiki/OpenCV> (open CV)
27. *Python*. Diversos autors. Modificat 2019. Viquipèdia. 30 de juny de 2020.
<https://ca.wikipedia.org/wiki/Python>
28. *PyCharm*. Diversos autors. Modificat 2020. Wikipedia. 2 de juliol de 2020.
<https://en.wikipedia.org/wiki/PyCharm>
29. *Llenguatge de programació*. Diversos autors. Modificat 2020. Viquipèdia. 3 de juliol de 2020.
https://ca.wikipedia.org/wiki/Llenguatge_de_programaci%C3%B3
30. *Entorn integrat de desenvolupament*. Diversos autors. Modificat 2020. Viquipèdia. 3 de juliol de 2020. https://ca.wikipedia.org/wiki/Entorn_integrat_de_desenvolupament
31. *Màquina virtual*. Diversos autors. Modificat 2019. Viquipèdia. 8 de juliol de 2020.
https://ca.wikipedia.org/wiki/M%C3%A0quina_virtual
32. *Transformaciones geométricas de imágenes*. Autor desconegut. Modificat 2018. Unipython. 8 de juliol de 2020. <https://unipython.com/transformaciones-geometricas-de-imagenes-con-opencv/>
33. *Sistema operatiu*. Diversos autors. Modificat 2020. Viquipèdia. 9 de juliol de 2020.
https://ca.wikipedia.org/wiki/Sistema_operatiu
34. *Definición de open source*. Julián Pérez Porto i Ana Gardey. Modificat 2014. Definición.DE. 9 de juliol de 2020. <https://definicion.de/open-source/>

35. *Desarrollo de programas android*. Diversos autors. Modificat 2020. Viquipèdia. 10 de juliol de 2020. https://es.wikipedia.org/wiki/Desarrollo_de_programas_para_Android#Kivy
36. *Librería informática, ¿De qué se trata?*. Alejandro Zorita. Modificat 2016. ByteCode. 10 de juliol de 2020. <https://bytecode.es/que-es-una-libreria-informatica/>
37. *Los lenguajes de programación más usados en la actualidad*. Autor desconegut. Modificat 2020. Universia. 10 de juliol de 2020. <https://noticias.universia.com.ar/consejos-profesionales/noticia/2016/02/22/1136443/conoce-cuales-lenguajes-programacion-populares.html>
38. *Librerías de visión artificial*. Autor desconegut. Modificat 2013. La mirada del Golem. 14 de juliol de 2020. <http://miradadelgolem.blogspot.com/2013/04/librerias-de-vision-artificial.html>
39. Figura portada. Imatge de l'autor (30-09-20).

ANNEXOS

COINTEUR



L'ÚS DE LA VISIÓ ARTIFICIAL PER AL RECOMPTE DE MONEDES

Anna Pruença i Mascort

TREBALL DE RECERCA 2N DE BAT

TUTORA: YOLANDA FORT

INS de CELRÀ (Celrà)

CURS 2020-2021

DATA DE PRESENTACIÓ: 14 d'octubre de 2020

1 ÍNDEX

1	ÍNDEX.....	1
2	ANNEX 1: Manual d'instal·lació del programari.....	2
3	ANNEX 2: Codi aplicació Python.....	20
4	ANNEX 3: Codi aplicació Kivy.....	36
5	ANNEX 4: Proves de l'aplicació.....	64

Les figures que hi ha a continuació no estan referenciades perquè pertanyen totes a l'autor del treball.

2 ANNEX 1: Manual d'instal·lació del programari

1. Instal·lació entorn la màquina virtual Virtualbox Oracle en Host

-Accedir a la secció de downloads de <https://www.virtualbox.org/>. La màquina HOST és la màquina en la qual s'instal·larà la màquina virtual. La màquina virtual permet treballar amb un sistema operatiu diferent a el de la màquina Host.

-Descarregar la versió Windows host i realitzar la instal·lació.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 5.2 packages, see [VirtualBox 5.2 builds](#). Please also use version supported until July 2020.

VirtualBox 6.1.0 platform packages

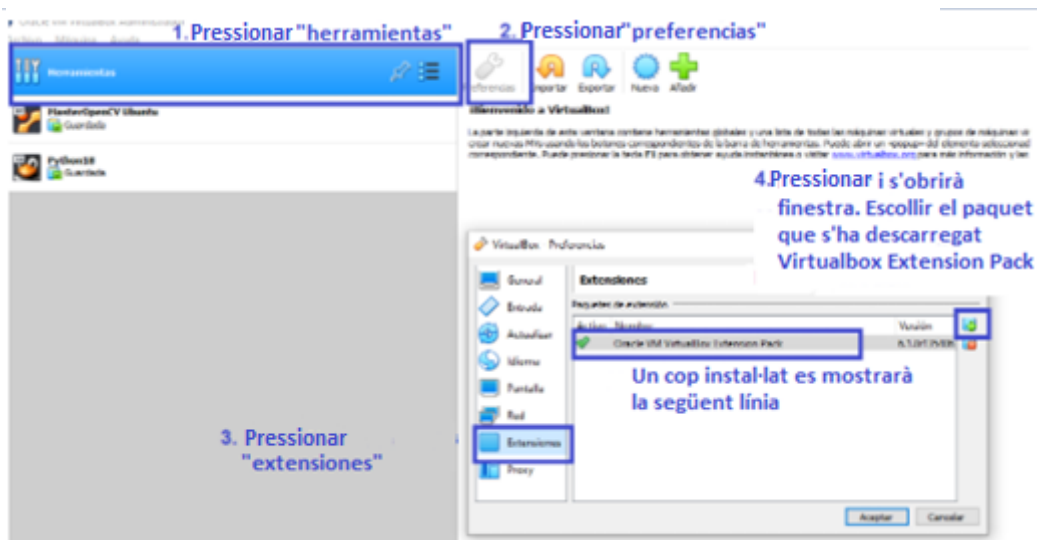
- [Windows hosts](#)
- [OS X hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)

-Un cop instal·lat, descarregar el paquet.

-Accedir a VirtualBox. Realitzar els 4 passos que es descriuen per a instal·lar el paquet Extension pack que ens donarà accés al control del port USB.

VirtualBox 6.1.0 Oracle VM VirtualBox Extension Pack

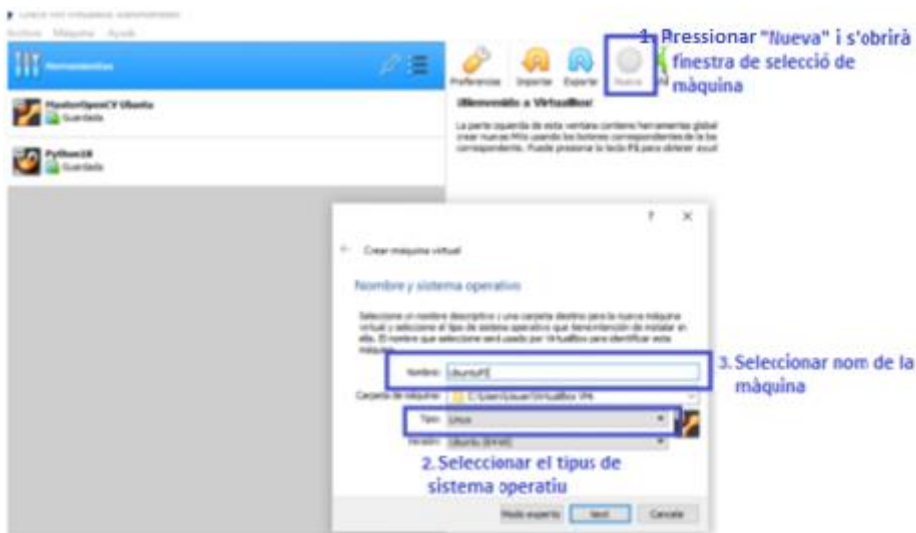
- [All supported platforms](#)



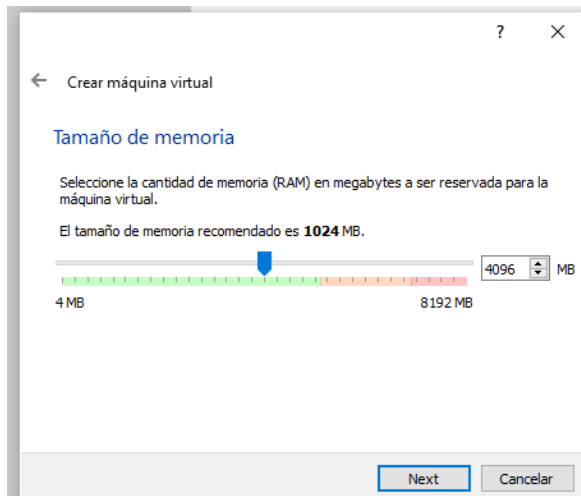
2. Instal·lació de la màquina virtual Ubuntu Guest

2.1 Instal·lació del hardware virtual Guest

-Un cop descarregada, obrir la màquina virtual per a procedir a la creació de la mateixa. En aquest procés només s'instal·la el "hardware de la màquina virtual". En un procés posterior s'instal·larà el software.



-Escriure un 50% de la memòria disponible del PC.



-Seleccionar "creación disco duro virtual" i pressionar "crear".


Disco duro

Si desea puede añadir un disco duro virtual a la nueva máquina. Puede crear un nuevo archivo de disco duro o seleccionar uno de la lista o de otra ubicación usando el icono de la carpeta.

. Si necesita una configuración de almacenamiento más compleja puede omitir este paso y hacer los cambios a las preferencias de la máquina virtual una vez creada.

El tamaño recomendado del disco duro es **10,00 GB**.

- No añadir un disco duro virtual
- Crear un disco duro virtual ahora
- Usar un archivo de disco duro virtual existente

MasterOpenCV Ubuntu-disk002.vdi (Normal, 30,80 GB) 

Crear

Cancelar

-Utilitzar la configuració que proposa el sistema y pressionar “Next”.

Tipo de archivo de disco duro

Seleccione el tipo de archivo que quiere usar para el nuevo disco duro virtual. Si no necesita usarlo con otro software de virtualización puede dejar esta configuración sin cambiar.

- VDI (VirtualBox Disk Image)
- VHD (Virtual Hard Disk)
- VMDK (Virtual Machine Disk)

Modo experto

Next

Cancelar

-Seleccionar la opción “tamaño fijo”.

Almacenamiento en unidad de disco duro física

Seleccione si el nuevo archivo de unidad de disco duro virtual debería crecer según se use (reserva dinámica) o si debería ser creado con su tamaño máximo (tamaño fijo).

Un archivo de disco duro **reservado dinámicamente** solo usará espacio en su disco físico a medida que se llena (hasta un máximo **tamaño fijo**), sin embargo no se reducirá de nuevo automáticamente cuando el espacio en él se libere.

Un archivo de disco duro de **tamaño fijo** puede tomar más tiempo para su creación en algunos sistemas, pero normalmente es más rápido al usarlo.

- Reservado dinámicamente
- Tamaño fijo


Next

Cancelar

-Seleccionar uns 20GB i pressionar “crear”.

Ubicación del archivo y tamaño

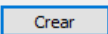
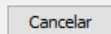
Escriba el nombre del archivo de unidad de disco duro virtual en el campo debajo o haga clic en el icono de carpeta para seleccionar una carpeta diferente donde crear el archivo.

C:\Users\Usuari\VirtualBox VMs\UbuntuPI\UbuntuPI.vdi 

Seleccione el tamaño de disco duro virtual en megabytes. Este tamaño es el límite para el archivo de datos que una máquina virtual podrá almacenar en el disco duro.

 20,76 GB

4,00 MB 2,00 TB

3. Instal·lació del software Guest

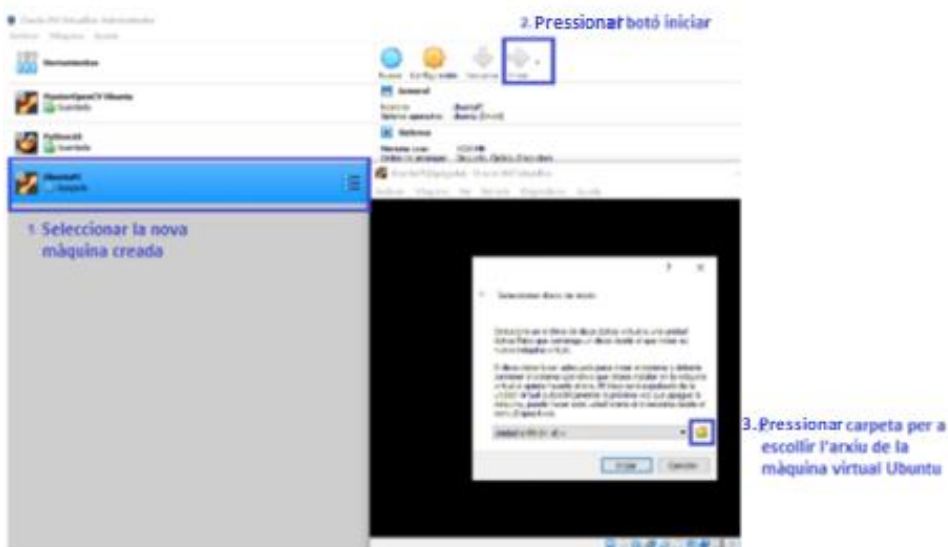
3.1 La descàrrega software màquina virtual Ubuntu

-Accedir a la secció download de la pàgina <https://ubuntu.com/>.

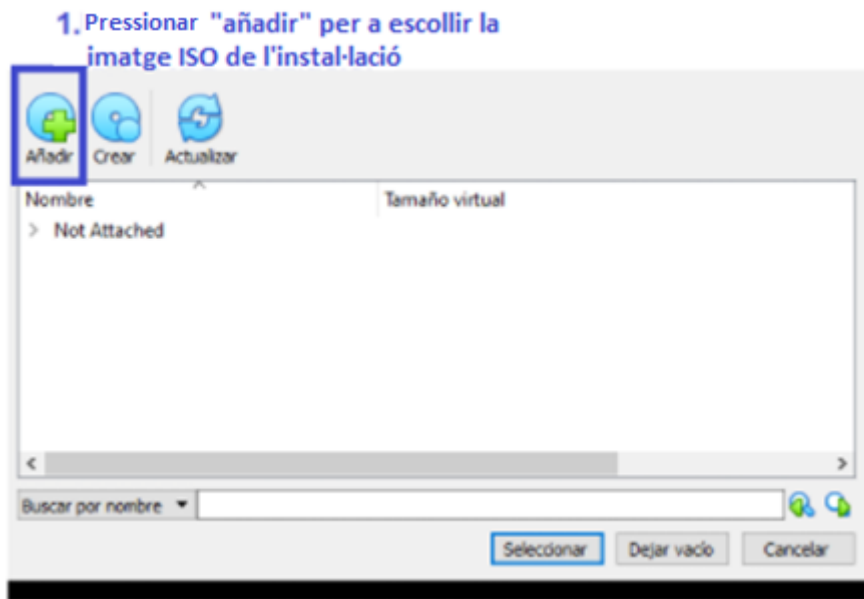
-En aquest cas es descarrega la imatge per a Ubuntu 18.04. Aquest arxiu s'ha baixat a D:\AI\Instalacion\ubuntu-18.04.3-desktop-amd64.iso. L'arxiu ISO d'instal·lació s'utilitzarà com a disc d'arrancament de l'instal·lació inicial de la màquina virtual.

4. Instal·lació del software màquina virtual

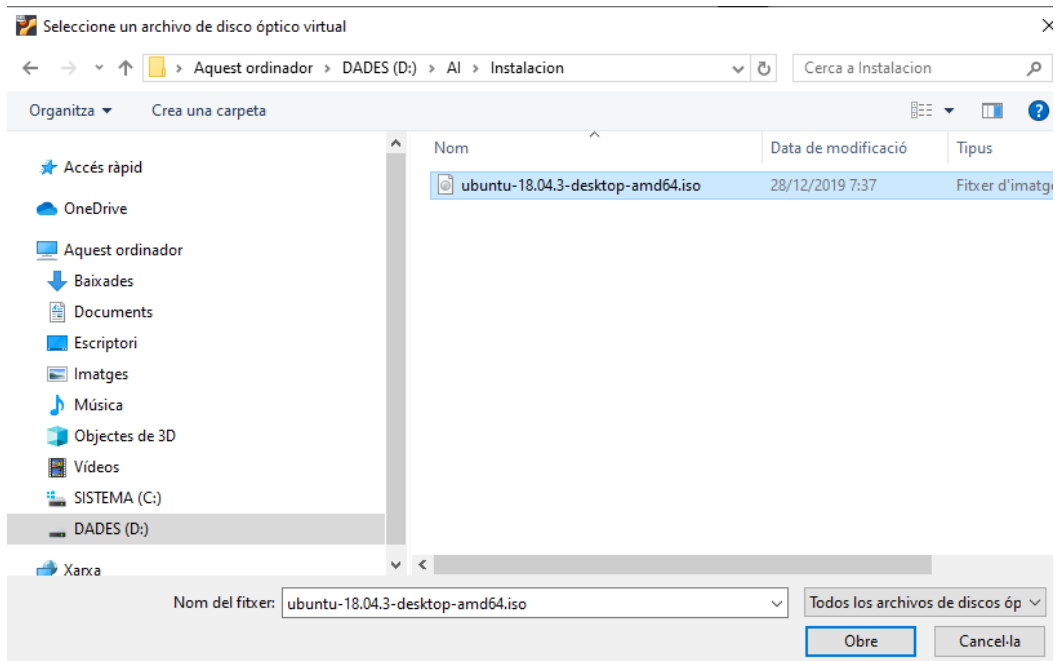
-Accedir a Virtualbox per a realitzar la instal·lació del software a la màquina Ubuntu.



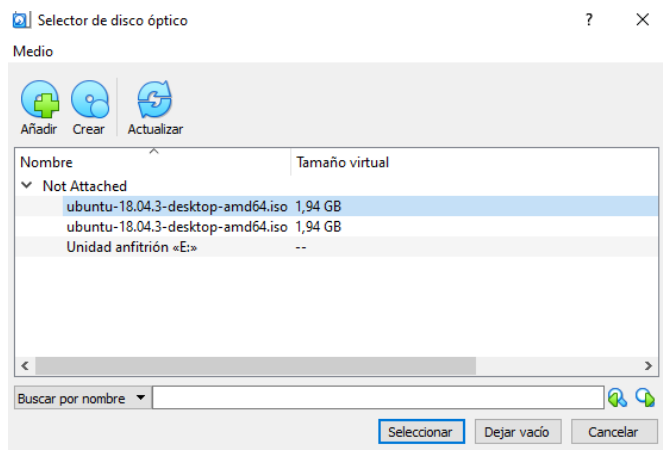
-Pressionar "añadir" per a triar la màquina ISO



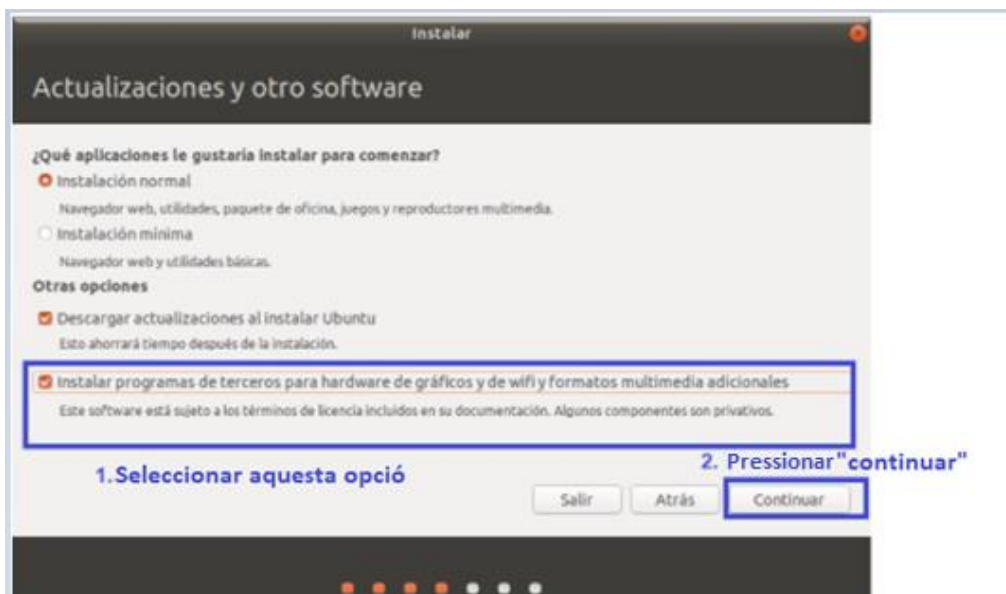
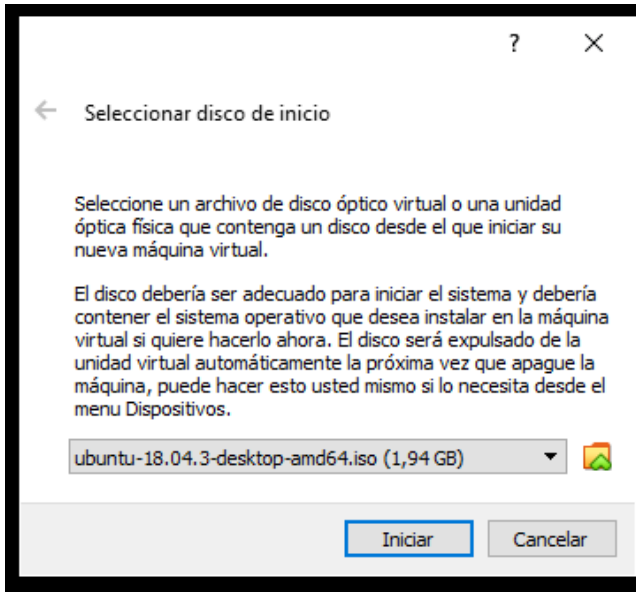
-S'ha de seleccionar l'arxiu de la imatge ISO descarregada a la secció 2.2.1 i pressionar obrir.

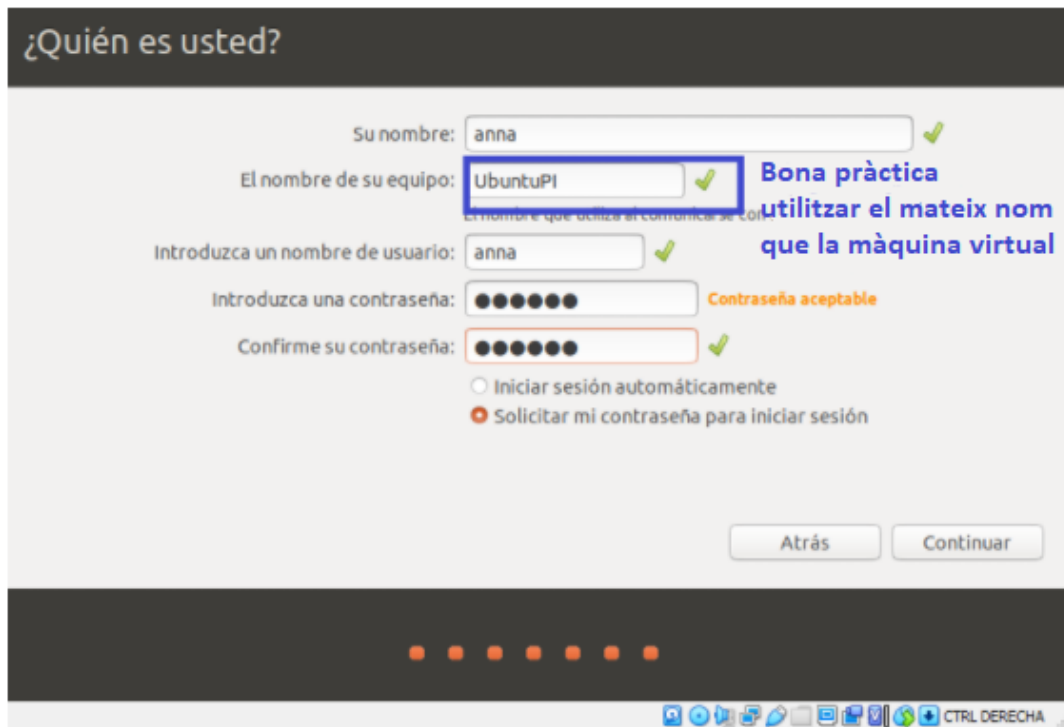


-Pressionar seleccionar.



-Pressionar Iniciar per a arrancar la màquina virtual

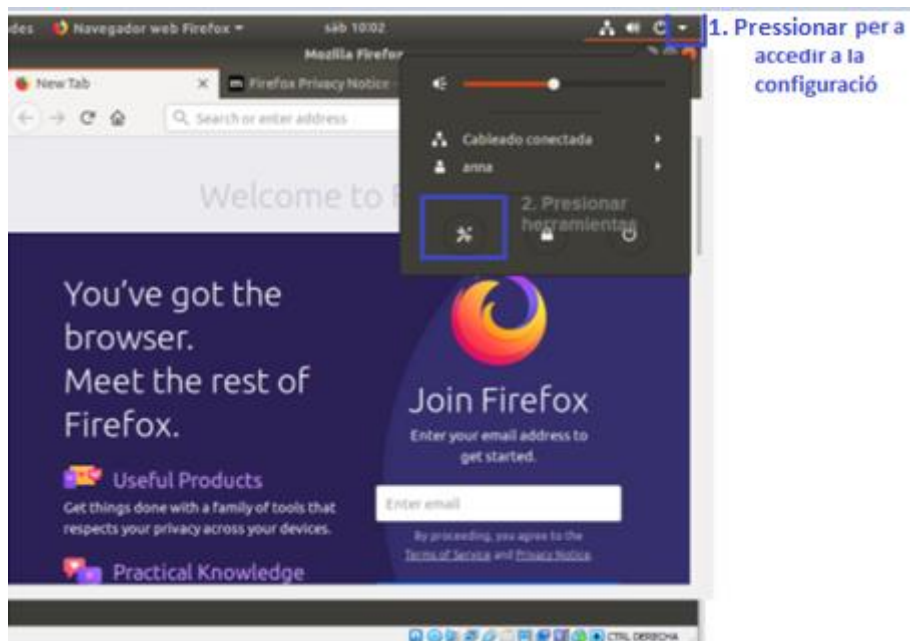




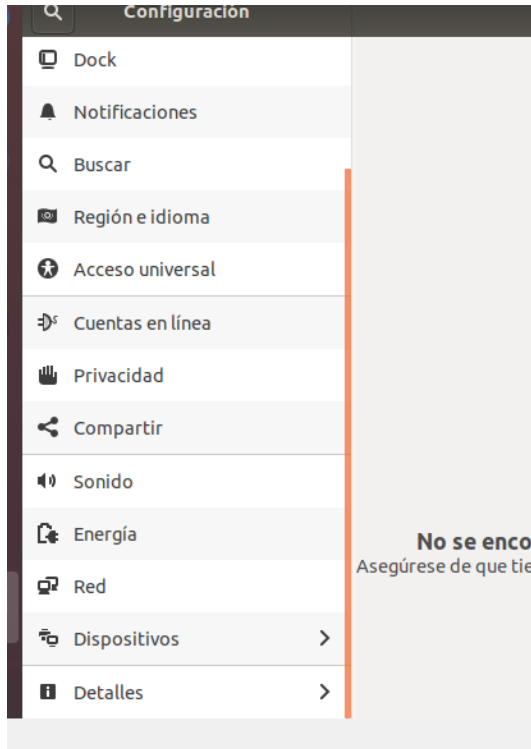
5. Configuració Ubuntu i connexió màquina Host.

5.1 Canvi resolució pantalla

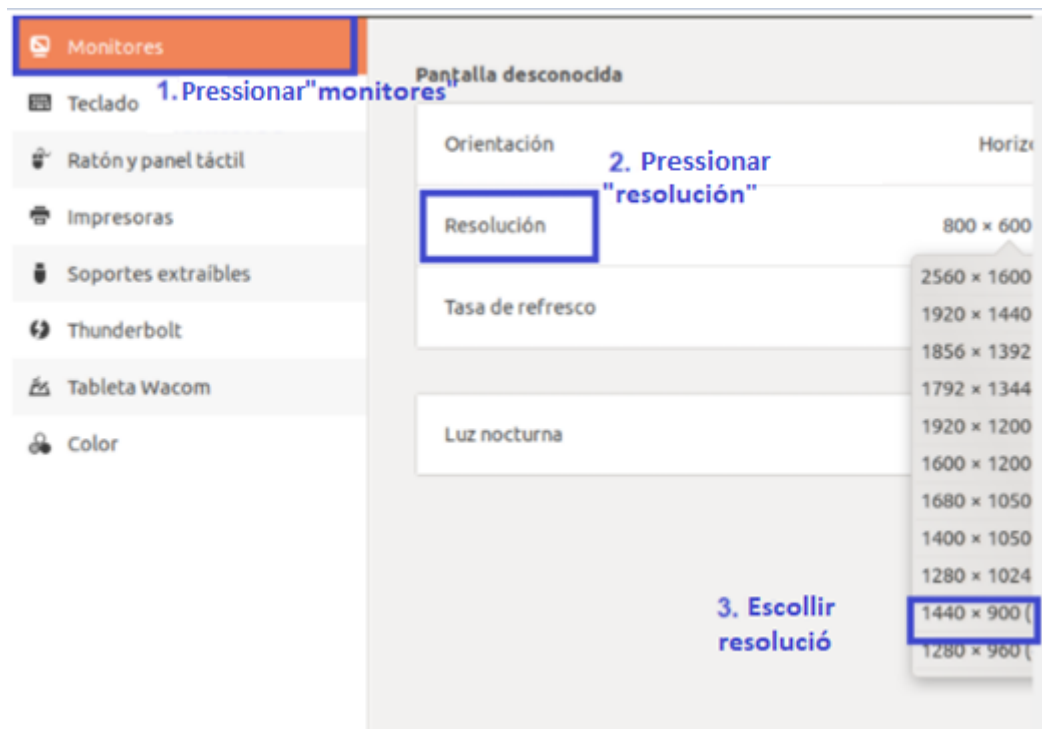
-Selecció de resolució de pantalla.



-Escollir dispositiu



-Es recomana la resolució 1440x900. Depèn del monitor.

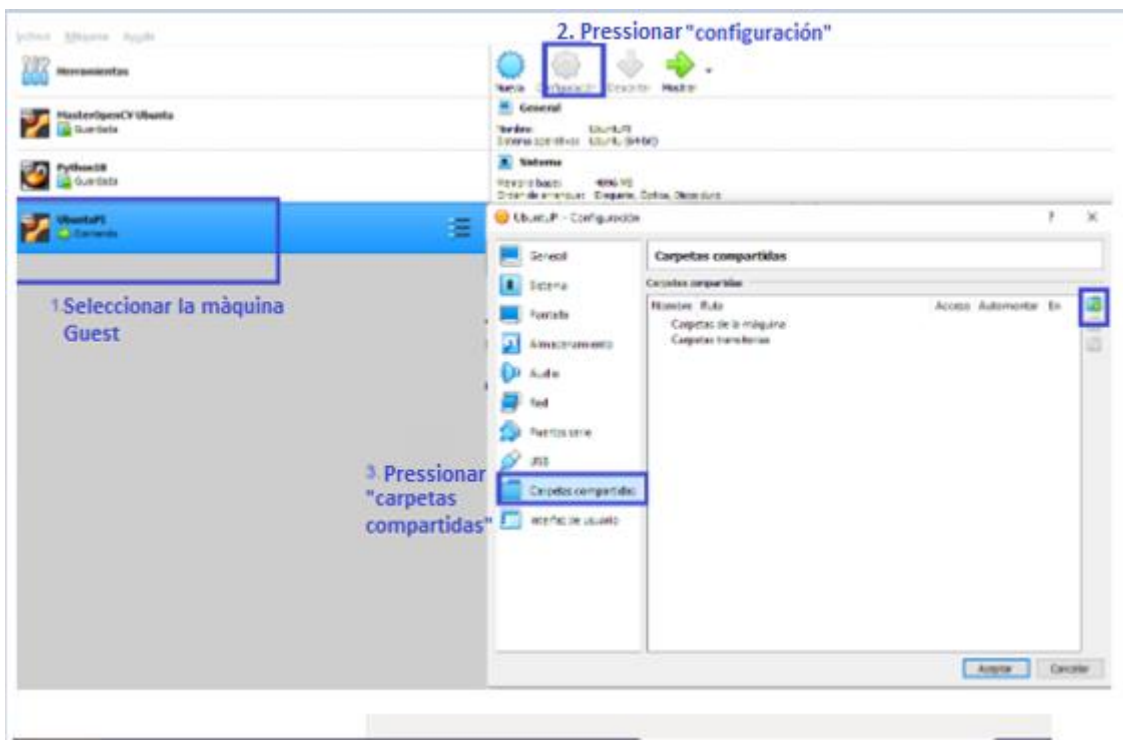


-Col·locar el ratolí sobre la finestra que s'ha obert i mantenir el botó premut per a arrastrar la pantalla fins que aparegui el botó aplicar.

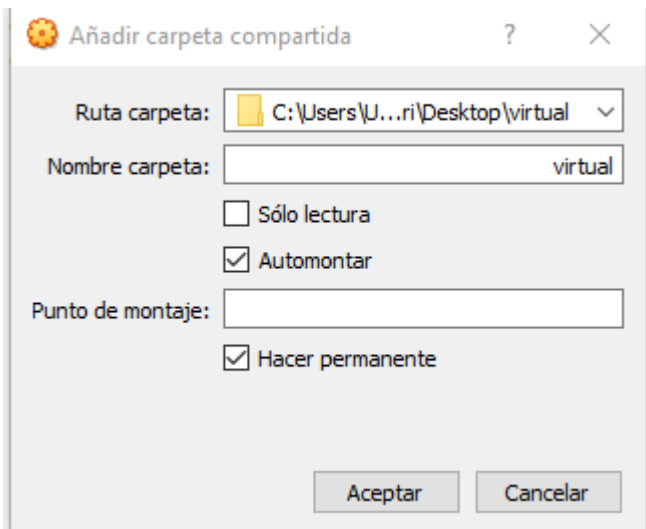


5.2 Compartir una carpeta entre màquina Host i Guest.

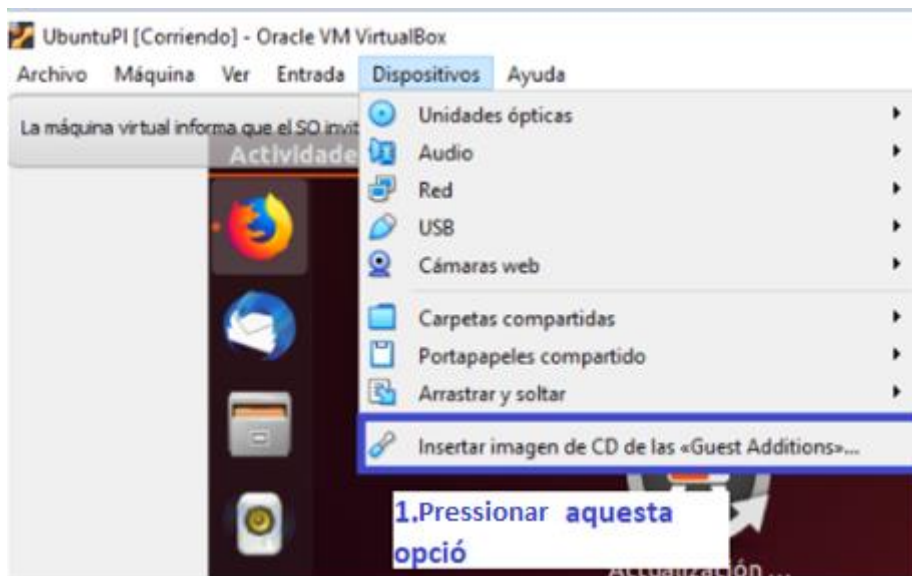
-Accedir a màquina virtual per a realitzar la configuració.



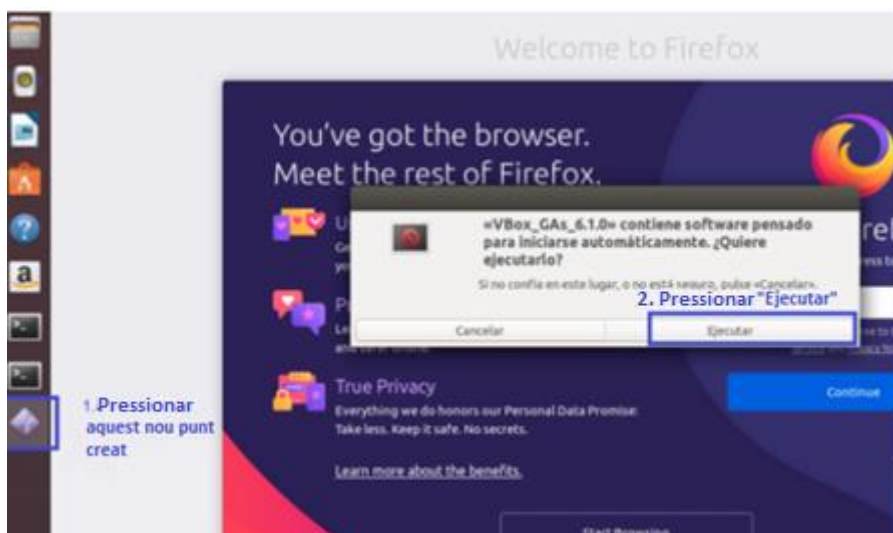
-Triar carpeta que es requereix compartir amb les opcions seleccionades. Reiniciar màquina Guest.



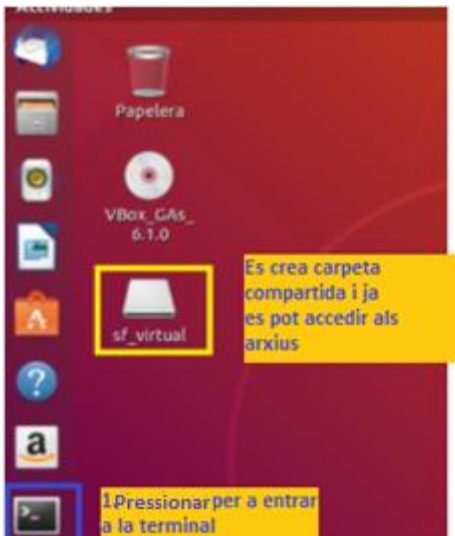
-Des de la màquina virtual instal·lar el CD de Guest Additions.



-Instal·lar les guest additions. Pressionar el nou botó creat.

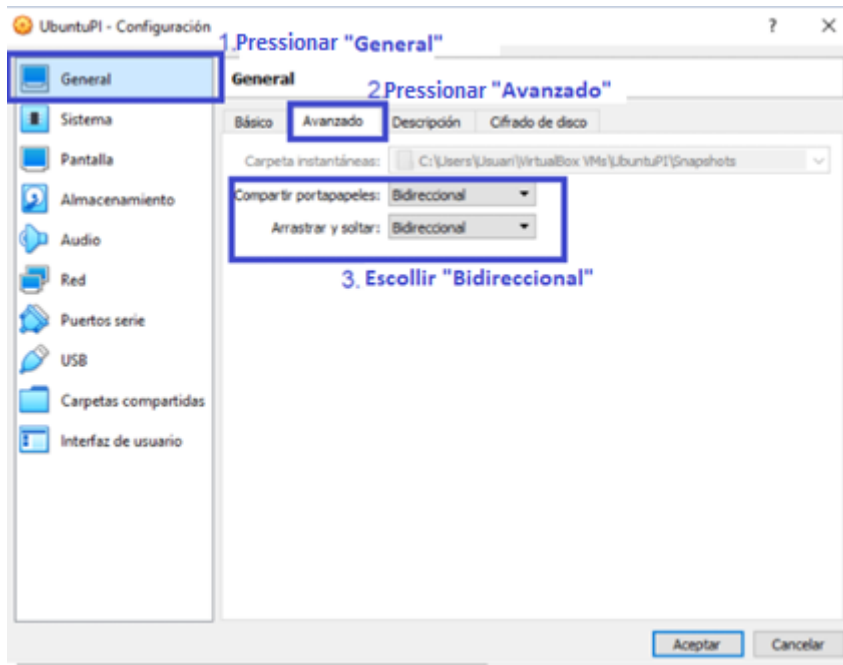


-Entrar a terminal per a enllaçar el host. Tot i que s'hagi creat carpeta encara no és possible accedir-hi.



5.3 Compartir portapapers Host i Guest

-Aquesta opció permet compartir portapapers entre la màquina Host i Guest.



5.4 Instal·lar Python

-Comandaments per a actualitzar els paquets d'Ubuntu (totes les línies que estan amb cursiva són instruccions de instal·lació que s'han de copiar a la terminal i donar a "enter"):

- *sudo apt-get update && sudo apt-get upgrade*

- `sudo apt update`

-Paquets per a instal·lador de Python:

- `sudo apt install python3-pip python-pip`

5.5 Instal·lar OpenCV

-Es segueixen les recomanacions de la pàgina <https://opencv.org/> per a la instal·lació amb Python:

- `sudo apt-get install python-opencv python3-opencv`
- `sudo apt-get install python-dev python-numpy python3-dev python3-numpy`
- `sudo apt-get install cmake gcc g++ libavcodec-dev libavformat-dev libswscale-dev`
- `sudo apt-get install libgstreamer-plugins-base1.0-dev libgstreamer1.0-dev libgtk2.0-dev libgtk-3-dev`

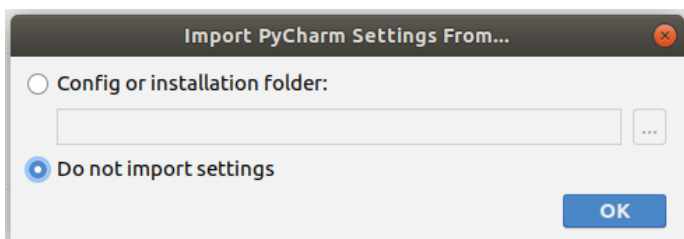
-Per a que reconegui diferents formats d'arxiu:

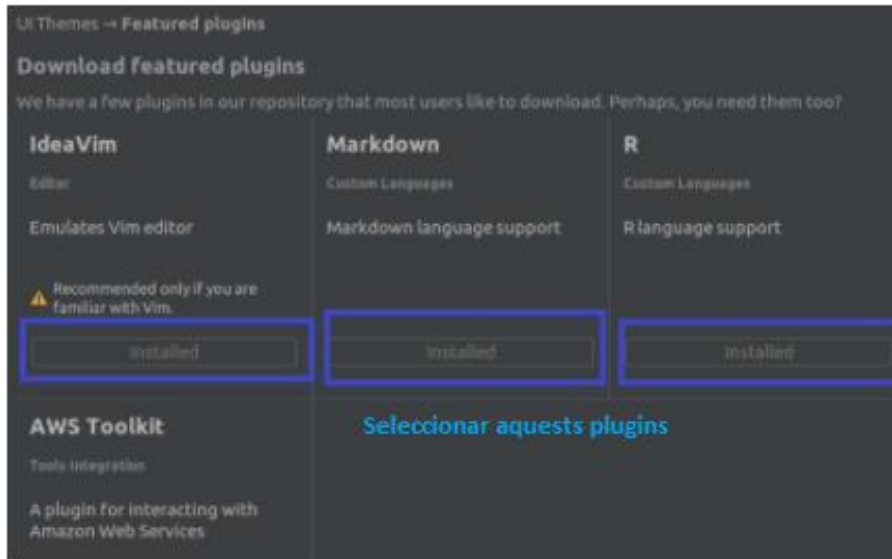
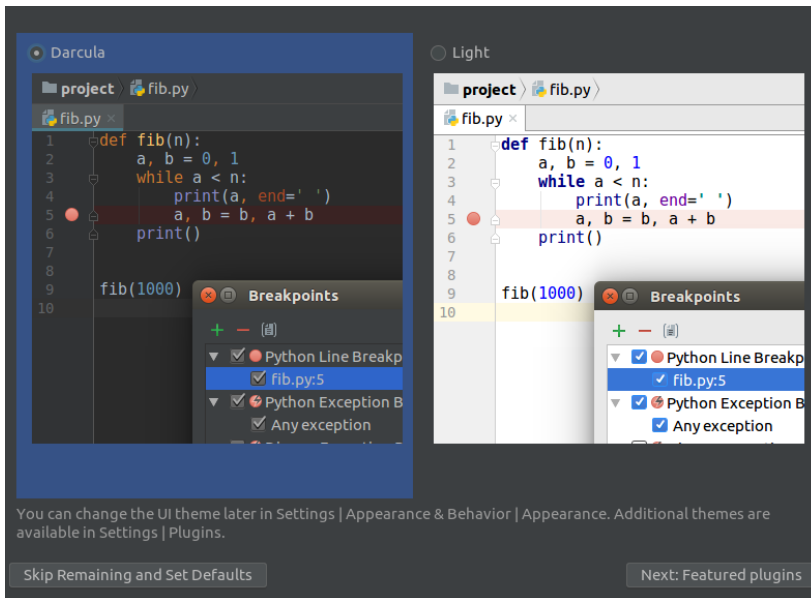
- `sudo apt-get install libpng-dev libjpeg-dev libopenexr-dev libtiff-dev libwebp-dev`

5.6 Instal·lar pycharm comunity edition

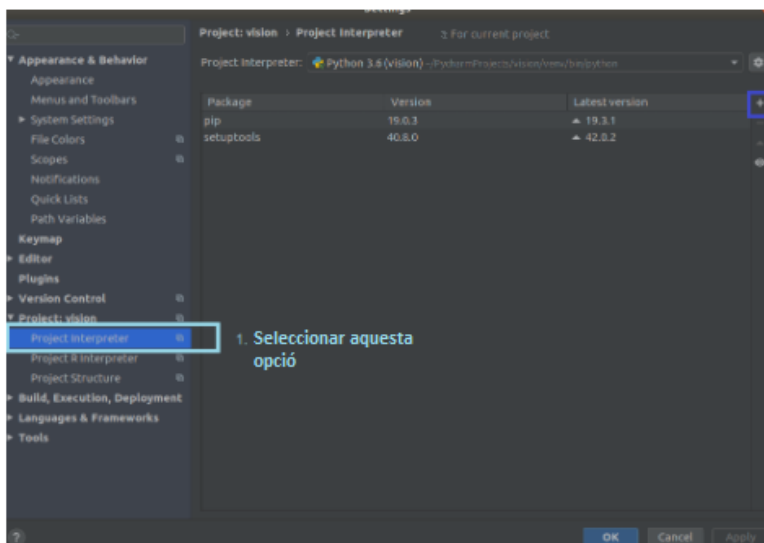
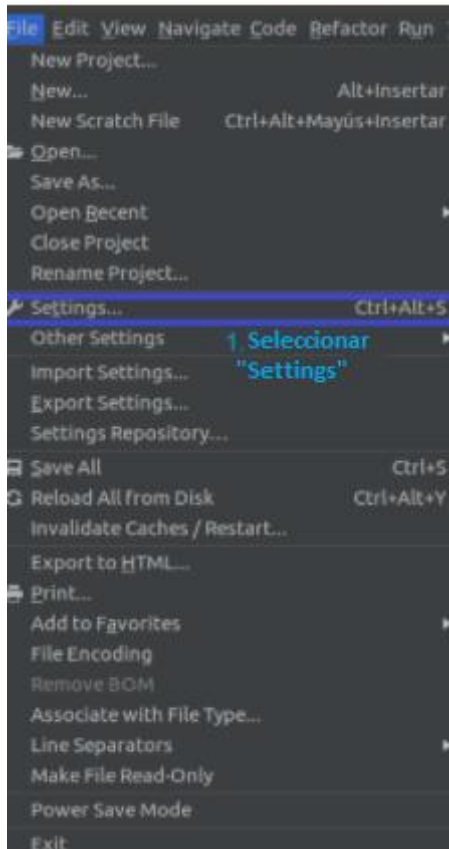
- `sudo snap install pycharm-community --classic.`

-Un cop instal·lat s'ha d'executar amb les següents opcions:

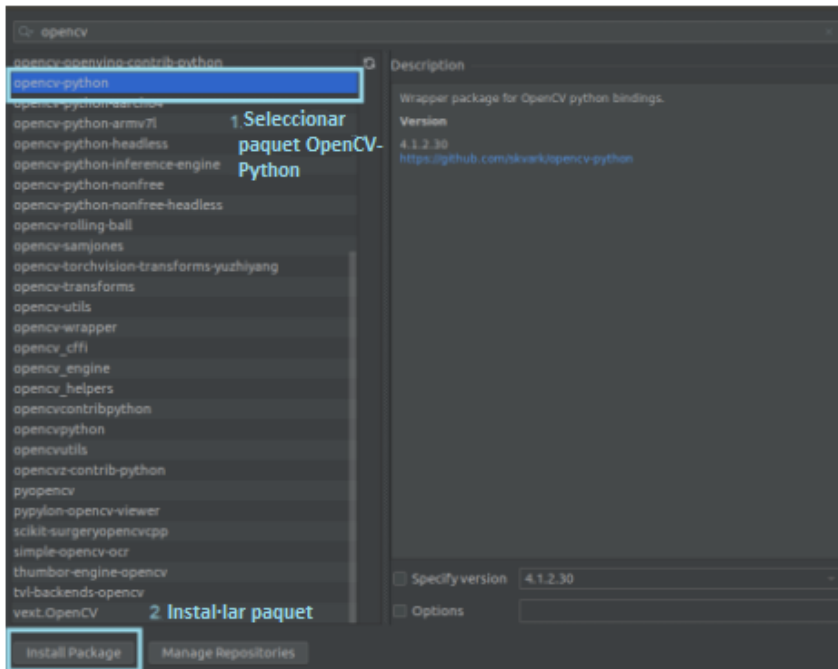




-Per a importar les llibreries d'Opencv, *numpy* s'ha de seleccionar la opció "Settings".



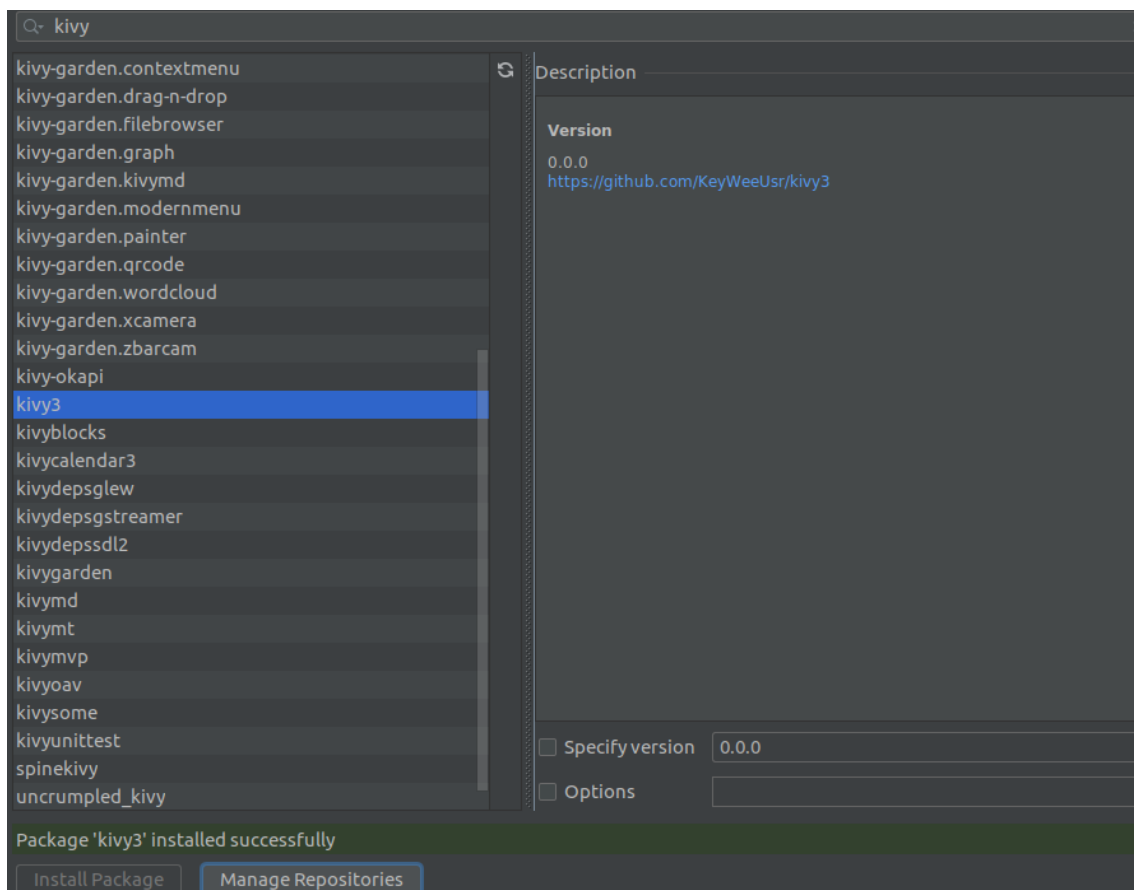
OPENCV



KIVY

- `sudo add-apt-repository ppa:kivy-team/kivy`
- `sudo apt-get update`
- `sudo apt-get install python3-kivy`

-Triar kivyblocks.



Actualització Master:

- *pip3 install notebook*
- *sudo python3 -m pip install -U matplotlib*

Càmera web

- *pip install opencv-contrib-python*
- *pycharm triar els contribution els free i non free.*

Executar

jupyter notebook

- *#!/bin/bash*
- *# Install Python pip*
- *sudo apt-get install -y curl*
- *sudo apt-get install -y python3-distutils*
- *curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py*
- *sudo python3 get-pip.py*

Dependencies with SDL2

Install necessary system packages

- *sudo apt-get install -y \
python-pip \
build-essential \
git \
python \
python3-dev \
ffmpeg \
libsdl2-dev \
libsdl2-image-dev \
libsdl2-mixer-dev \
libsdl2-ttf-dev \
libportmidi-dev \
libswscale-dev \
libavformat-dev \
libavcodec-dev *

- *zlib1g-dev*

Dependencies Kivy

- *sudo pip3 install cython*

Install Kivy

- *sudo pip3 install kivy*

Dependencies Buildozer

- *sudo apt install -y *
- *build-essential *
- *ccache *
- *git *
- *libncurses5:i386 *
- *libstdc++6:i386 *
- *libgtk2.0-0:i386 *
- *libpangox-1.0-0:i386 *
- *libpangoxft-1.0-0:i386 *
- *libidn11:i386 *
- *python2.7 *
- *python2.7-dev *
- *openjdk-8-jdk *
- *unzip *
- *zlib1g-dev *
- *zlib1g:i386 *
- *libltdl-dev *
- *libffi-dev *
- *libssl-dev *
- *libtool *
- *autoconf *
- *automake *
- *pkgconf *

- *autotools-dev* \
- *cmake*

Install Buildozer

Cloning Buildozer repository directly to the HOME directory causes the

AttributeError 'Namespace' object has no attribute 'ignore_setup_py' error when trying to build apk file.

So Buildozer repository should be cloned to any directory except HOME

- *mkdir ~/buildozer-repo*
- *cd ~/buildozer-repo*
- *git clone https://github.com/kivy/buildozer.git*
- *cd buildozer*
- *sudo python3 setup.py install*
- *""""*
- *buildozer init*
- *buildozer android debug*

3 ANNEX 2: Codi aplicació Python

```

import cv2
from kivy.uix.button import Button
from kivy.uix.gridlayout import GridLayout
from kivy.uix.textinput import TextInput
from kivy.uix.label import Label
from kivy.app import App
from kivy.uix.filechooser import FileChooserListView
from kivy.uix.widget import Widget
from kivy.uix.image import Image
from kivy.properties import ObjectProperty
import numpy as np
from kivy.uix.checkbox import CheckBox
from kivy.uix.popup import Popup
from kivy.uix.slider import Slider
from kivy.uix.spinner import Spinner
from kivy.uix.tabbedpanel import TabbedPanel, TabbedPanelItem, TabbedPanelContent
from kivy.uix.switch import Switch
from kivy.uix.screenmanager import Screen, ScreenManager
import speech_recognition as sr
from gtts import gTTS
import os
import time
import playsound
global exec
global archive
archive=""

exec="0"

class ScreenManagement (ScreenManager):
    pass

class Aplicacio2App(App):
    def build (self):
        return ScreenManagement ()
class Screen1 (Screen):
    #self.im.reload()

    #executar canals

    def on_pre_enter(self, *args):
        global exec
        if exec==0:
            print("first")
        if exec==1:
            self.im.source = "fotofinalr.jpg"
            self.im.reload()

            self.textim.text = archive

#carregar arxiu
def fca (self):
    #self.im.source=self.textim.text
    #imatgeinicial=cv2.imread(self.textim.text)
    #cv2.imwrite("fotofinalr.jpg",imatgeinicial)
    #self.im.reload()
    pass

```

```

#guardar imatge
def fgi (self):
    nomarxiufinal=self.textimf.text

    imatgeinicial = cv2.imread("fotofinalr.jpg")

    cv2.imwrite(nomarxiufinal,imatgeinicial)

#textimf

## En pop UP
def fdesenfocament (self):
    odes= desenfocament()
    odes.open()
    odes.imd.reload()

def fcam (self):
    cap = cv2.VideoCapture(0)

    while (True):
        # Capture frame-by-frame
        ret, frame = cap.read()

        # Our operations on the frame come here
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        # Display the resulting frame
        cv2.imshow('frame', gray)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            cv2.imwrite("fotofinalr.jpg", gray)
            break

    # When everything done, release the capture

    cap.release()
    cv2.destroyAllWindows()
    self.im.source="fotofinalr.jpg"
    self.im.reload()
class load(Screen):

# pathh=StringProperty(os.getcwd())
# print(pathh.defaultvalue)
# print("3")
def fload(self, path, filename):
    #print("fil", filename,"path",path)

    fotorot = cv2.imread(filename[0])
    cv2.imwrite("fotofinalr.jpg", fotorot)
    global exec
    global archive
    exec=1
    archive=filename[0]
    self.manager.current = "Screen_1"

class Canals (Screen):

```



```

def on_pre_enter(self, *args):
    self.imc.source = "fotofinalr.jpg"
    self.imc.reload()

fotor = cv2.imread("fotofinalr.jpg")

def fbc (self, tabuc,vermell, verd, blau, filtrer, filtreg, filtreb,
filtreh, filtres, filtrev ,control):
    #print(tabuc,vermell, verd, blau, filtrer, filtreg, filtreb,
control)

fotor = cv2.imread("fotofinalr.jpg")
alcada, amplada = fotor.shape[:2]

B,G,R= cv2.split(fotor)
zeros = np.zeros(fotor.shape[:2], dtype="uint8")

one= np.ones (fotor.shape[:2], dtype="uint8")
#Corregir fotor per fotorot
if tabuc=="RGB":
    if filtrer==True and filtreb==False and filtreg==False:
        fotor = cv2.merge([zeros, zeros, R])
    if filtrer == True and filtreb == True and filtreg == False:
        fotor = cv2.merge([B, zeros, R])
    if filtrer == True and filtreb == True and filtreg == True:
        fotor = cv2.merge([B, G, R])
    if filtrer == True and filtreb == False and filtreg == True:
        fotor = cv2.merge([zeros, G, R])
    if filtrer == False and filtreb == True and filtreg == True:
        fotor = cv2.merge([B, G, zeros])
    if filtrer == False and filtreb == True and filtreg == False:
        fotor = cv2.merge([B, zeros, zeros])
    if filtrer == False and filtreb == False and filtreg == True:
        fotor = cv2.merge([zeros, G, zeros])
    if filtrer == False and filtreb == False and filtreg == False:
        fotor = cv2.merge([zeros, zeros, zeros])
    B, G, R = cv2.split(fotor)
    if filtrer==True:
        if int(vermell) > 0:
            R = cv2.add(R, one * int(vermell))
        if int(vermell) < 0:
            RR = int(vermell) * -1 * one
            R = cv2.subtract(R, RR)

    if filtreb == True:
        if int(blau) > 0:
            B = cv2.add(B, one * int(blau))
        if int(blau) < 0:
            BB = int(blau) * -1 * one
            B = cv2.subtract(B, BB)
    if filtreg == True:
        if int(verd) > 0:
            G = cv2.add(G, one * int(verd))
        if int(verd) < 0:
            GG = int(verd) * -1 * one
            G = cv2.subtract(G, GG)
    fotor =cv2.merge ([B,G,R])
if tabuc == "B/N":
    fotor = cv2.cvtColor(fotor, cv2.COLOR_BGR2GRAY)
if tabuc == "HSV":
    fotor = cv2.cvtColor(fotor, cv2.COLOR_BGR2HSV)
    H, S, V = cv2.split(fotor)
    if filtreh==True and filtres==False and filtrev==False:

```

```

        fotor=cv2.merge([H,zeros, zeros])
    if filtreh == True and filtres == True and filtrev == False:
        fotor = cv2.merge([H, S, zeros])
    if filtreh == True and filtres == True and filtrev == True:
        fotor = cv2.merge([H, S, V])
    if filtreh == True and filtres == False and filtrev == True:
        fotor = cv2.merge([H, zeros, V])
    if filtreh == False and filtres == True and filtrev == False:
        fotor = cv2.merge([zeros, S, zeros])
    if filtreh == False and filtres == True and filtrev == True:
        fotor = cv2.merge([zeros, S, V])
    if filtreh == False and filtres == False and filtrev == True:
        fotor = cv2.merge([zeros, zeros, V])
    if filtreh == False and filtres == False and filtrev == False:
        fotor = cv2.merge([zeros, zeros, zeros])

cv2.imwrite("fotofinalp.jpg",fotor)
self.imc.source="fotofinalp.jpg"
self.imc.reload()
if control=="a":
    cv2.imwrite ("fotofinalr.jpg",fotor)
    global exec
    exec = 1
    self.manager.current = "Screen_1"

def fcancelar (self):
    self.manager.current = "Screen_1"
class rotacio (Screen):

def on_pre_enter(self, *args):
    self.imr.source = "fotofinalr.jpg"
    self.imr.reload()

def fcancelar (self):
    self.manager.current = "Screen_1"
def fgr(self, angle, flipv, fliph, control, escala):

    fotorot = cv2.imread("fotofinalr.jpg")
    alcada, amplada = fotorot.shape[:2]

    rotation_matrix = cv2.getRotationMatrix2D((int(amplada/2),
int(alcada/2)), int(angle), float(escala))
    rotated_image = cv2.warpAffine (fotorot, rotation_matrix,
(int(amplada), int(alcada)))

    if fliph==True:
        rotated_image=cv2.flip(rotated_image,1)
    if flipv==True:
        rotated_image=cv2.flip(rotated_image,0)

    if control=="a":
        cv2.imwrite ("fotofinalr.jpg", rotated_image)
        global exec
        exec = 1
        self.manager.current = "Screen_1"
    else:
        cv2.imwrite("fotofinalp.jpg", rotated_image)
        self.imr.source="fotofinalp.jpg"
        self.imr.reload ()

class desenfocament (Screen):

```

```

fotor = cv2.imread("fotofinalr.jpg")
def on_pre_enter(self, *args):
    self.imd.source = "fotofinalr.jpg"
    self.imd.reload()
def fcancelar (self):
    self.manager.current = "Screen_1"
def fdesenfocament (self,tabu,median, gauss, radi, sigmac, sigmax,
control ):
    fotor = cv2.imread("fotofinalr.jpg")
    if tabu == "Median":
        fotor = cv2.imread("fotofinalr.jpg")
        blur = cv2.blur(fotor, (int(median), int(median)))

    if tabu == "Gauss":
        fotor = cv2.imread("fotofinalr.jpg")
        #perquè és 0 l'últim?
        blur = cv2.GaussianBlur(fotor, (int(gauss), int(gauss)),0)

    if tabu == "Bilateral":
        fotor = cv2.imread("fotofinalr.jpg")
        #perquè és 0 l'últim?
        blur = cv2.bilateralFilter(fotor,
int(radi),int(sigmac),int(sigmax))

    if control=="a":
        cv2.imwrite ("fotofinalr.jpg",blur)
        global exec
        exec = 1
        self.manager.current = "Screen_1"
    else:
        cv2.imwrite("fotofinalp.jpg", blur)
        self.imd.source = "fotofinalp.jpg"
        self.imd.reload()
class umbral (Screen):
    fotor = cv2.imread("fotofinalr.jpg")
    def on_pre_enter(self, *args):
        self.imu.source = "fotofinalr.jpg"
        self.imu.reload()
    def fcancelar (self):
        self.manager.current = "Screen_1"
    def fu (self, umbral , control):
        fotor = cv2.imread("fotofinalr.jpg")
        alcada, amplada = fotor.shape[:2]
        #print (umbral,control)
        ret, thresh1 = cv2.threshold(fotor, int(umbral), 255,
cv2.THRESH_BINARY)

    if control=="a":
        cv2.imwrite("fotofinalr.jpg",thresh1)
        global exec
        exec = 1
        self.manager.current = "Screen_1"
    else:
        cv2.imwrite("fotofinalp.jpg", thresh1)
        self.imu.source = "fotofinalp.jpg"
        self.imu.reload()
class erosio (Screen):
    fotor = cv2.imread("fotofinalr.jpg")
    def on_pre_enter(self, *args):
        self.ime.source = "fotofinalr.jpg"
        self.ime.reload()

```

```

def fed (self, erosio, sic1,
sed1,sv1,sic2,sed2,sv2,sic3,sed3,sv3,control):

    erosion = cv2.imread("fotofinalr.jpg")
    alcada, amplada = erosion.shape[:2]

    # Let's define our kernel size
    kernel = np.ones((int(erosio), int(erosio)), np.uint8)

    # Now we erode

#True
if sed1=="Erosió" and sic1==True:
    erosion = cv2.erode(erosion, kernel, iterations=int(sv1))
if sed1 == "Dilatació" and sic1 == True:
    erosion = cv2.dilate(erosion, kernel, iterations=int(sv1))

if sed2 == "Erosió" and sic2 == True:
    erosion = cv2.erode(erosion, kernel, iterations=int(sv2))
if sed2 == "Dilatació" and sic2 == True:
    erosion = cv2.dilate(erosion, kernel, iterations=int(sv2))
if sed3 == "Erosió" and sic3 == True:
    erosion = cv2.erode(erosion, kernel, iterations=int(sv3))
if sed3 == "Dilatació" and sic3 == True:
    erosion = cv2.dilate(erosion, kernel, iterations=int(sv3))

if control=="p":
    cv2.imwrite("fotofinalp.jpg",erosion)
    self.ime.source="fotofinalp.jpg"
    self.ime.reload()

if control == "a":
    cv2.imwrite("fotofinalr.jpg", erosion)
    global exec
    exec = 1
    self.manager.current = "Screen_1"

def fcancelar (self):
    self.manager.current = "Screen_1"
class enfoc(Screen):
    fotor = cv2.imread("fotofinalr.jpg")
    def on_pre_enter(self, *args):
        self.imef.source = "fotofinalr.jpg"
        self.imef.reload()
    def fenfoc (self, spv, control):

        if spv== "Enfocament":
            fotor = cv2.imread("fotofinalr.jpg")
            alcada, amplada = fotor.shape[:2]

            # Create our shapening kernel, we don't normalize since the
            # the values in the matrix sum to 1
            kernel_sharpening = np.array([[ -1,  -1,  -1],
                                           [-1,  9,  -1],
                                           [-1,  -1,  -1]])

            # applying different kernels to the input image
            Enfocat = cv2.filter2D(fotor, -1, kernel_sharpening)
        if spv=="Pas alt":
            fotor = cv2.imread("fotofinalr.jpg")
            alcada, amplada = fotor.shape[:2]

```

```

    kernel_highpass = np.array([[0, -1, 0], [-1, 5, -1], [0, -1,
0]])
    Enfocat = cv2.filter2D(fotor, -1, kernel_highpass)

    if control=="p":
        cv2.imwrite("fotofinalp.jpg", Enfocat)
        self.imef.source = "fotofinalp.jpg"
        self.imef.reload()

    if control=="a":
        cv2.imwrite("fotofinalr.jpg",Enfocat)
        global exec
        exec = 1
        self.manager.current = "Screen_1"
#nomarxiufinal=self.textimf.text

    def fcancelar(self):
        self.manager.current = "Screen_1"
class costats (Screen):
    fotor = cv2.imread("fotofinalr.jpg")
    def on_pre_enter(self, *args):
        self.imc.source = "fotofinalr.jpg"
        self.imc.reload()
    def fcostats(self, tabuc, gradm, gradd, sobelm, cx, cy, laplaces, control,):

        fotor = cv2.imread("fotofinalr.jpg")
        alcada, amplada = fotor.shape[:2]
        costats=fotor

        if tabuc=="Canny":

            costats = cv2.Canny(fotor, int(gradd),int(gradm))
        if tabuc=="Sobel":

            # Extract Sobel Edges
            if cx==True and cy==True:

                sobel_x = cv2.Sobel(fotor, cv2.CV_64F, 0, 1,
ksize=int(sobelm))

                sobel_y = cv2.Sobel(fotor, cv2.CV_64F, 1, 0,
ksize=int(sobelm))

            # suma dos imagenes
            costats = cv2.bitwise_or(sobel_x, sobel_y)
            if cx==True and cy==False:
                fotor = cv2.imread("fotofinalr.jpg")
                alcada, amplada = fotor.shape[:2]
                costats=cv2.Sobel(fotor, cv2.CV_64F, 0, 1,
ksize=int(sobelm))
            if cy==True and cx==False:
                fotor = cv2.imread("fotofinalr.jpg")
                alcada, amplada = fotor.shape[:2]
                costats=cv2.Sobel(fotor, cv2.CV_64F, 1, 0,
ksize=int(sobelm))

        if tabuc=="Laplace":
            fotor = cv2.imread("fotofinalr.jpg")
            alcada, amplada = fotor.shape[:2]
            costats = cv2.Laplacian(fotor, cv2.CV_64F, ksize=int(laplaces))

```



```

    master = [1, 0.887, 0.815, 0.747, 0.681, 0.588, 0.530, 0.398],
[1.5, 1, 0.919, 0.842, 0.768, 0.663, 0.598, 0.449], [1.5, 1.5, 1, 0.916,
0.835, 0.722, 0.650, 0.488], [1.5, 1.5, 1.5, 1, 0.912, 0.788, 0.710, 0.533],
[1.5, 1.5, 1.5, 1.5, 1, 0.864, 0.779, 0.585], [1.5, 1.5, 1.5, 1.5, 1.5, 1,
0.901, 0.677], [1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1, 0.751], [1.5, 1.5, 1.5, 1.5,
1.5, 1.5, 1.5, 1.5, 1]
    ratiose = []
    # monedes = ["2 EUR", "1 EUR", "50 ct"]
    monedes = ["2EUR", "50ct", "1EUR", "20ct", "5ct", "10ct",
"2ct", "1ct"]
    #per detectar quina és la línia màster que s'ha d'utilitzar
    monedareferencia = [0]
    for (i, aa) in enumerate(monedes):
        if moneda == aa:
            l = i
            monedareferencia = monedes[l]
    #print("Posicio Moneda", l)
    c = 0
    for i in range(len(sorted_contours) - 1):
        if cv2.contourArea(sorted_contours[c]) >= int(area):
            c = i + 1
            b = cv2.contourArea(sorted_contours[c]) /
cv2.contourArea(sorted_contours[0])
            ratiose.append(b)

    #print(ratiose)
    zz = []
    ratiosef=[]
    errorr=[]
    ratiotrobat = 0

    resultats = [0, 0, 0, 0, 0, 0, 0, 0]
    tol = 0.04
    for (i, c) in enumerate(range(len(ratiose))):
        ##for l in range(len(master)):
        for z in range(len(master[0])):

            if (ratiose[c] >= (master[l][z]) * (1 - tol) and
ratiose[c] <= (master[l][z]) * (1 + tol)) and (
                ratiose[c] < (1 - tol) or ratiose[c] > (1 +
tol)):

                idcountor[c + 1] = c + 1
                idarea[c + 1] = monedes[z]
                #print("Moneda referencia", monedes[l], "moneda a
determinar", monedes[z])

                zz.append(monedes[z])
                ratiotrobat = 1
                ratiotrobat = 1
                ratiosef.append(ratiose[c])
                errorr.append(abs((master[l][z] - ratiose[c]) /
master[l][z] * 100))
                if ratiose[c] >= (1 - tol) or ratiose[c] <= (1 + tol):
                    ratiotrobat=1

    if ratiotrobat == 1:
        for q in zz:
            if q == "2EUR":
                resultats[0] = resultats[0] + 1
            if q == "1EUR":
                resultats[1] = resultats[1] + 1
            if q == "50ct":
                resultats[2] = resultats[2] + 1

```

```

if q == "20ct":
    resultats[3] = resultats[3] + 1
if q == "10ct":
    resultats[4] = resultats[4] + 1
if q == "5ct":
    resultats[5] = resultats[5] + 1
if q == "2ct":
    resultats[6] = resultats[6] + 1
if q == "1ct":
    resultats[7] = resultats[7] + 1
for (i, y) in enumerate(ratiose):
if y >= 1 - tol and y <= 1 + tol:
    if monedareferencia == "2EUR":
        resultats[0] = resultats[0] + 1
        idcountor[i + 1] = i + 1
        idarea[i + 1] = "2EUR"
    if monedareferencia == "1EUR":
        resultats[1] = resultats[1] + 1
        idcountor[i + 1] = i + 1
        idarea[i + 1] = "1EUR"
    if monedareferencia == "50ct":
        resultats[2] = resultats[2] + 1
        idcountor[i + 1] = i + 1
        idarea[i + 1] = "50ct"
    if monedareferencia == "20ct":
        resultats[3] = resultats[3] + 1
        idcountor[i + 1] = i + 1
        idarea[i + 1] = "20ct"
    if monedareferencia == "10ct":
        resultats[4] = resultats[4] + 1
        idcountor[i + 1] = i + 1
        idarea[i + 1] = "10ct"
    if monedareferencia == "5ct":
        resultats[5] = resultats[5] + 1
        idcountor[i + 1] = i + 1
        idarea[i + 1] = "5ct"
    if monedareferencia == "2ct":
        resultats[6] = resultats[6] + 1
        idcountor[i + 1] = i + 1
        idarea[i + 1] = "2ct"
    if monedareferencia == "1ct":
        resultats[7] = resultats[7] + 1
        idcountor[i + 1] = i + 1
        idarea[i + 1] = "1ct"
if monedareferencia == "2EUR":
    resultats[0] = resultats[0] + 1
    idcountor[0] = 0
    idarea[0] = "2EUR"
if monedareferencia == "1EUR":
    resultats[1] = resultats[1] + 1
    idcountor[0] = 0
    idarea[0] = "1EUR"
if monedareferencia == "50ct":
    resultats[2] = resultats[2] + 1
    idcountor[0] = 0
    idarea[0] = "50ct"
if monedareferencia == "20ct":
    resultats[3] = resultats[3] + 1
    idcountor[0] = 0
    idarea[0] = "20ct"
if monedareferencia == "10ct":
    resultats[4] = resultats[4] + 1
    idcountor[0] = 0
    idarea[0] = "10ct"

```



```

if monedareferencia == "5ct":
    resultats[5] = resultats[5] + 1
    idcontador[0] = 0
    idarea[0] = "5ct"
if monedareferencia == "2ct":
    resultats[6] = resultats[6] + 1
    idcontador[0] = 0
    idarea[0] = "2ct"
if monedareferencia == "1ct":
    resultats[7] = resultats[7] + 1
    idcontador[0] = 0
    idarea[0] = "1ct"
#print(resultats)
ncont = 0

for (i, c) in enumerate(sorted_contours):

    if cv2.contourArea(c) > int(area):
        if int(alcada * 4 / 2832) < 1:
            fescala = 1
        else:
            fescala = int(alcada * 4 / 2832)
        CM = cv2.moments(c)
        cx = int(CM['m10'] / CM['m00'])
        cy = int(CM['m01'] / CM['m00'])
        if idarea[i]!="0":
            cv2.circle(ones, (cx, cy), fescala*4, (0, 0,
255), -1)
            cv2.drawContours(ones, [c], -1, (0, 255, 0),
fescala*4)
            cv2.putText(ones, str(idarea[i]), (cx, cy),
cv2.FONT_HERSHEY_SIMPLEX, fescala, (30, 200, 100), fescala*2)
            ncont = ncont + 1

x = 0
a = 0
b = 0
c = 0
d = 0
e = 0
f = 0
g = 0
h = 0

for o in range(len(idarea)):
    if idarea[o] == "2EUR":
        x = x + 2
        a = a + 1
    if idarea[o] == "1EUR":
        x = x + 1
        b = b + 1
    if idarea[o] == "50ct":
        x = x + 0.50
        c = c + 1
    if idarea[o] == "20ct":
        x = x + 0.20
        d = d + 1
    if idarea[o] == "10ct":
        x = x + 0.10
        e = e + 1
    if idarea[o] == "5ct":
        x = x + 0.05
        f = f + 1
    if idarea[o] == "2ct":

```



```

0.710,0.533], [1.5,1.5,1.5,1.5,1,0.864,0.779,0.585], [1.5, 1.5, 1.5, 1.5,
1.5, 1, 0.901, 0.677], [1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1, 0.751], [1.5,
1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1]
    ratiose = []
    # monedes = ["2 EUR", "1 EUR", "50 ct"]
    monedes = ["2EUR", "50ct", "1EUR", "20ct", "5ct", "10ct",
"2ct", "1ct"]
    monedareferencia = [0]
    for (i, aa) in enumerate(monedes):
        if moneda == aa:
            l = i
            monedareferencia = monedes[l]
    #print("Posicio Moneda", l)
    c = 0
    for i in range(len(sorted_contours) - 1):
        if cv2.contourArea(sorted_contours[c]) >= int(area):
            c = i + 1
            b = cv2.contourArea(sorted_contours[c]) /
cv2.contourArea(sorted_contours[0])
            ratiose.append(b)

    #print(ratiose)
    zz = []
    ratiotrobat = 0

    resultats = [0, 0, 0, 0, 0, 0, 0, 0]
    tol = 0.04
    ratiosef=[]
    errorr=[]
    for (i, c) in enumerate(range(len(ratiose))):
        ##for l in range(len(master)):
        for z in range(len(master[0])):
            if (ratiose[c] >= (master[l][z]) * (1 - tol) and
ratiose[c] <= (master[l][z]) * (1 + tol)) and (
                ratiose[c] < (1 - tol) or ratiose[c] > (1 +
tol)):
                    idcounter[c + 1] = c + 1
                    idarea[c + 1] = monedes[z]
                    #print("Moneda referencia", monedes[l], "moneda a
determinar", monedes[z])

                    zz.append(monedes[z])
                    ratiotrobat = 1
                    ratiosef.append(ratiose[c])
                    errorr.append(abs((master[l][z]-
ratiose[c])/master[l][z]*100))
                    if ratiose[c] >= (1 - tol) or ratiose[c] <= (1 + tol):
                        ratiotrobat=1

    #print(zz)
    if ratiotrobat == 1:
        for q in zz:
            if q == "2EUR":
                resultats[0] = resultats[0] + 1
            if q == "1EUR":
                resultats[1] = resultats[1] + 1
            if q == "50ct":
                resultats[2] = resultats[2] + 1
            if q == "20ct":
                resultats[3] = resultats[3] + 1
            if q == "10ct":
                resultats[4] = resultats[4] + 1
            if q == "5ct":
                resultats[5] = resultats[5] + 1
            if q == "2ct":

```

```

        resultats[6] = resultats[6] + 1
    if q == "1ct":
        resultats[7] = resultats[7] + 1
for (i, y) in enumerate(ratiose):
    if y >= 1 - tol and y <= 1 + tol:
        if monedareferencia == "2EUR":
            resultats[0] = resultats[0] + 1
            idcountor[i + 1] = i + 1
            idarea[i + 1] = "2EUR"
        if monedareferencia == "1EUR":
            resultats[1] = resultats[1] + 1
            idcountor[i + 1] = i + 1
            idarea[i + 1] = "1EUR"
        if monedareferencia == "50ct":
            resultats[2] = resultats[2] + 1
            idcountor[i + 1] = i + 1
            idarea[i + 1] = "50ct"
        if monedareferencia == "20ct":
            resultats[3] = resultats[3] + 1
            idcountor[i + 1] = i + 1
            idarea[i + 1] = "20ct"
        if monedareferencia == "10ct":
            resultats[4] = resultats[4] + 1
            idcountor[i + 1] = i + 1
            idarea[i + 1] = "10ct"
        if monedareferencia == "5ct":
            resultats[5] = resultats[5] + 1
            idcountor[i + 1] = i + 1
            idarea[i + 1] = "5ct"
        if monedareferencia == "2ct":
            resultats[6] = resultats[6] + 1
            idcountor[i + 1] = i + 1
            idarea[i + 1] = "2ct"
        if monedareferencia == "1ct":
            resultats[7] = resultats[7] + 1
            idcountor[i + 1] = i + 1
            idarea[i + 1] = "1ct"
    if monedareferencia == "2EUR":
        resultats[0] = resultats[0] + 1
        idcountor[0] = 0
        idarea[0] = "2EUR"
    if monedareferencia == "1EUR":
        resultats[1] = resultats[1] + 1
        idcountor[0] = 0
        idarea[0] = "1EUR"
    if monedareferencia == "50ct":
        resultats[2] = resultats[2] + 1
        idcountor[0] = 0
        idarea[0] = "50ct"
    if monedareferencia == "20ct":
        resultats[3] = resultats[3] + 1
        idcountor[0] = 0
        idarea[0] = "20ct"
    if monedareferencia == "10ct":
        resultats[4] = resultats[4] + 1
        idcountor[0] = 0
        idarea[0] = "10ct"
    if monedareferencia == "5ct":
        resultats[5] = resultats[5] + 1
        idcountor[0] = 0
        idarea[0] = "5ct"
    if monedareferencia == "2ct":
        resultats[6] = resultats[6] + 1
        idcountor[0] = 0

```

```

        idarea[0] = "2ct"
if monedareferencia == "1ct":
        resultats[7] = resultats[7] + 1
        idcounter[0] = 0
        idarea[0] = "1ct"
#print(resultats)

ncont = 0
#print("Hola" ,alcada)
for (i, c) in enumerate(sorted_contours):

    if cv2.contourArea(c) > int(area):
        if int(alcada * 4 / 2832) < 1:
            fescala = 1
        else:
            fescala = int(alcada * 4 / 2832)
        CM = cv2.moments(c)
        cx = int(CM['m10'] / CM['m00'])
        cy = int(CM['m01'] / CM['m00'])
        if idarea[i]!="0":
            cv2.circle(ones, (cx, cy), fescala*4, (0, 0,
255), -1)

            cv2.drawContours(ones, [c], -1, (0, 255, 0),
fescala*4)

            cv2.putText(ones, str(idarea[i]), (cx, cy),
cv2.FONT_HERSHEY_SIMPLEX, fescala, (0, 255, 0), fescala*2)
            ncont = ncont + 1

#for (i, c) in enumerate(sorted_contours):
#if idarea[0]=="0":
#fescala = int(alcada * 4 / 2832)
#print ("HOLA")
#noshadet = ["no s'ha detectat cap moneda"]
#CM = cv2.moments(c)
#cx = int(CM['m10'] / CM['m00'])
#cy = int(CM['m01'] / CM['m00'])
#cv2.putText(ones, str(noshadet[0]), (cx, cy),
cv2.FONT_HERSHEY_SIMPLEX, fescala, (0, 255, 0),fescala * 2)

x = 0
a = 0
b = 0
c = 0
d = 0
e = 0
f = 0
g = 0
h = 0

for o in range(len(idarea)):
    if idarea[o] == "2EUR":
        x = x + 2
        a = a + 1
    if idarea[o] == "1EUR":
        x = x + 1
        b = b + 1
    if idarea[o] == "50ct":
        x = x + 0.50
        c = c + 1
    if idarea[o] == "20ct":
        x = x + 0.20
        d = d + 1
    if idarea[o] == "10ct":
        x = x + 0.10

```

```

        e = e + 1
    if idarea[o] == "5ct":
        x = x + 0.05
        f = f + 1
    if idarea[o] == "2ct":
        x = x + 0.02
        g = g + 1
    if idarea[o] == "1ct":
        x = x + 0.01
        h = h + 1

sumatotal = str(round(x, 2))
speak("tens" + str(sumatotal) + "euros")
ratiosed=[ '%.3f' % elem for elem in ratiosef]
errorf = ['%.1f' % elem for elem in errorf]
#print (errorf)
#self.ids.dades.text="Proporcions Area:" + str(ratiosed)
#self.ids.errorl.text="Errors monedes"+ str(errorf)
self.ids.sumatotal.text = str(round(x,2))
self.ids.dos.text = str(a)
self.ids.u.text = str(b)
self.ids.cinquanta.text = str(c)
self.ids.vint.text = str(d)
self.ids.deu.text = str(e)
self.ids.cinc.text = str(f)
self.ids.dosc.text = str(g)
self.ids.uc.text = str(h)
cv2.imwrite("fotofinalp.jpg", oness)
self.imm.source = "fotofinalp.jpg"
self.imm.reload()

def fcancelar(self):
    self.manager.current = "Screen_1"

Aplicacio2App().run()

```

4 ANNEX 3: Codi aplicació Kivy

<CustomLabel@Label>:

font_size:"20dp"

bold:True

color: 0.5,0.80,0.9,1

<CustomSpinner@Spinner>:

font_size:"16dp"

bold:True

color: 0.5,0.80,0.9,1

<CustomButton@Button>:

font_size:"20dp"

bold:True

color: 0.5,0.80,0.9,1

<CustomCheckBox@CheckBox>:

font_size:"16dp"

color: 0.5,0.80,0.9,1

<ScreenManagement>:

Screen1:

id:Screen1

Canals:

id:Canals

rotacio:

id:rotacio

desenfocament:

id:desenfocament

enfoc:

id: enfoc

umbral:

id:umbral

costats:

id:costats

erosio:

id:erosio

monedes:

id:monedes

load:

id:load

<Screen1>:

name:"Screen_1"

#im=nom d'imatge del gridlayour

im:im

#textim=nom d'imatge que s'ha dobrir

textim:textim

#guardar imatge (el nom)

textimf:textimf

#textinput red

GridLayout:

cols:1

size:root.width,root.height

size_hint:1,1

GridLayout:

cols:3

#Dades

GridLayout:

size_hint_x:0.2

cols:1

CustomLabel:

size_hint_y:0.05

CustomLabel:

size_hint_y:0.05

text:"Obre arxiu"

CustomLabel:

size_hint_y:0.05

text:".jpg"

id:textim

CustomButton:

text:"Cercar Imatge"

on_release:root.manager.current = "load_1"

size_hint_y:0.05

CustomLabel:

text:"Guardar arxiu"

size_hint_y:0.05

TextInput:

multiline:False

```
id:textimf
```

```
size_hint_y:0.05
```

```
text: ".jpg"
```

```
CustomButton:
```

```
text:"Acceptar"
```

```
on_press:root.fgi()
```

```
size_hint_y:0.05
```

```
Label:
```

```
size_hint_y:0.2
```

```
CustomButton:
```

```
text: "obrir càmera"
```

```
on_press: root.fcam()
```

```
size_hint_y: 0.2
```

```
#IMATGE
```

```
GridLayout:
```

```
cols:1
```

```
size_hint_x:0.6
```

```
Image:
```

```
id:im
```

```
#Retocs
```

```
GridLayout:
```

```
cols:1
```

```
size_hint_x:0.2
```

```
CustomButton:
```

```
text: "Canals"
```

```
on_release:root.manager.current = "Canals_1"
```

```
CustomButton:
```

```
text:"Rotació"
```

```
on_release:root.manager.current = "rotacio_1"
```

```
CustomButton:
```

```
text:"Desenfocamentament"
```

```
on_release:root.manager.current = "desenfocament_1"
```

```
CustomButton:
```

```
text: "Llindar"
```

```
on_release:root.manager.current = "umbral_1"
```

```
CustomButton:
```

```
text: "Erosió i dilatació"
```

```
on_release:root.manager.current = "erosio_1"
```

```
CustomButton:
```

```
text:"Enfocament"
```

```
on_release:root.manager.current = "enfoc_1"
```

```
CustomButton:
```

```
text: "Costats"
```

```
on_release:root.manager.current = "costats_1"
```

```
CustomButton:
```

```
text: "Monedes"
```

```
on_release:root.manager.current = "monedes_1"
```

```
<load>:
```

```
#: import o os
```

```
name:'load_1'
```

```
BoxLayout:
```

```
size: root.size
```

```
pos: root.pos
```

```
orientation: "vertical"
```

```
FileChooserListView:
```

```
path:o.getcwd()
```

id: filechooser

BoxLayout:

size_hint_y: None

height: 30

Button:

text: "Cancel·lar"

on_press: root.manager.current="Screen_1"

Button:

text: "Load"

on_press: root.fload(filechooser.path, filechooser.selection)

<Canals>:

imc:imc

name:"Canals_1"

GridLayout:

cols:1

size:root.width,root.height

size_hint:1,1

GridLayout:

cols:2

Image:

size_hint_x:0.55

source: "fotofinalr.jpg"

id:imc

GridLayout:

cols:1

size_hint_x:0.45

TabbedPanel:

id:tabuc

do_default_tab: False

TabbedPanelItem:

text:"RGB"

GridLayout:

cols:4

CheckBox:

active: True

id:filtrer

CustomLabel:

text: "vermell"

Slider:

min:-255

max:255

step:1

id:vermellx

CustomLabel:

text:str(int(vermellx.value))

id: vermell

CheckBox:

id:filtreg

active: True

CustomLabel:

text: "verd"

Slider:

min:-255

max:255

step:1

id:verdx

CustomLabel:

text:str(int(verdx.value))

id: verd

CheckBox:

id:filtreb

active: True

CustomLabel:

text: "blau"

Slider:

min:-255

max:255

step:1

id:blaux

CustomLabel:

text:str(int(blaux.value))

id: blau

TabbedPanellItem:

text: "B/N"

TabbedPanellItem:

text: "HSV"

GridLayout:

cols:2

#CheckBox:

#group: "filtreRGB"

#id:filtrer

CheckBox:

id:filtreh

active:True

CustomLabel:

```
text: "H"
```

```
CheckBox:
```

```
id:filtres
```

```
active:True
```

```
CustomLabel:
```

```
text: "S"
```

```
CheckBox:
```

```
id:filtrev
```

```
active:True
```

```
CustomLabel:
```

```
text: "V"
```

```
GridLayout:
```

```
cols:3
```

```
size_hint_y:0.3
```

```
CustomButton:
```

```
text: "Previsualització"
```

```
on_press: root.fbc(tabuc.current_tab.text,vermell.text,verd.text,blau.text, filtrer.active,
filtreg.active,filtreb.active,filtreh.active, filtros.active, filtrev.active, "p")
```

```
CustomButton:
```

```
text:"Acceptar"
```

```
on_press: root.fbc(tabuc.current_tab.text,vermell.text,verd.text,blau.text, filtrer.active,
filtreg.active,filtreb.active, filtreh.active, filtros.active, filtrev.active, "a")
```

```
CustomButton:
```

```
text: "Cancel·lar"
```

```
on_press: root.fcancelar()
```

<rotacio>:

name:"rotacio_1"

title: "Rotació"

imr:imr

GridLayout:

cols:1

size:root.width,root.height

size_hint:1,1

GridLayout:

cols:2

Image:

source: "fotofinalr.jpg"

id:imr

size_hint_x:0.55

GridLayout:

cols:1

size_hint_x: 0.45

GridLayout:

cols:3

CustomLabel:

text:"Angle"

size_hint_x:0.01

Slider:

min:-360

max:360

step:1

id:anglee

size_hint_x:0.5

CustomLabel:

text:str(int(anglee.value))

id:angle

size_hint_x:0.12

CustomLabel:

text:"Escala"

Slider:

min:0.1

max:1

value:1

step:0.01

id:escalaa

CustomLabel:

text:str(float(escalaa.value))

id:escala

GridLayout:

cols:2

CustomLabel:

text:"Invertir vertical"

CheckBox:

id:flipv

CustomLabel:

text:"Invertir horitzontal"

CheckBox:

id:fliph

CustomLabel:

GridLayout:

padding:2

cols: 3

CustomLabel:

CustomLabel:

CustomLabel:

CustomButton:

text:"Previsualització"

on_press: root.fgr (angle.text, fliph.active, flipv.active,"p",escala.text)

CustomButton:

text: "Acceptar"

on_press: root.fgr (angle.text, fliph.active, flipv.active,"a",escala.text)

CustomButton:

text: "Cancel·lar"

on_press: root.fcancelar()

<desenfocament>:

imd:imd

title: "desenfocament"

name: "desenfocament_1"

GridLayout:

cols:1

size:root.width,root.height

size_hint:1,1

GridLayout:

cols:2

Image:

```
size_hint_x:0.55
source: "fotofinalr.jpg"
id:imd
```

GridLayout:

```
cols:1
size_hint_x:0.45
```

TabbedPanel:

```
id:tabu
do_default_tab: False
```

TabbedPanelItem:

```
text: "Median"
```

GridLayout:

```
rows:1
CustomLabel:
    text: "Matriu"
CustomSpinner:
    text:"3"
    values:["3","5","7","9","11"]
    id:median
```

TabbedPanelItem:

```
text: "Gauss"
```

GridLayout:

```
rows:1
CustomLabel:
    text: "Matriu"
CustomSpinner:
    text:"3"
    values:["3","5","7","9","11"]
```

id:gauss

TabbedPanelItem:

text: "Bilateral"

GridLayout:

cols:3

CustomLabel:

text:"Radi"

Slider:

min:0

max:100

step:1

id:radix

value:9

CustomLabel:

text:str(int(radix.value))

id:radi

CustomLabel:

text:"SigmaC"

Slider:

min:0

max:100

step:1

id:sigmacx

value:75

CustomLabel:

text:str(int(sigmacx.value))

id:sigmac

CustomLabel:

text:"Sigma X"

Slider:

min:0

max:100

step:1

id:sigma α

value:75

CustomLabel:

text:str(int(sigma α .value))

id:sigma α

CustomLabel:

CustomLabel:

GridLayout:

cols:1

GridLayout:

cols:3

CustomButton:

text:"Previsualitzar"

on_press: root.fdesenfocament (tabu.current_tab.text,median.text, gauss.text, radi.text, sigmac.text, sigma α .text,"p")

CustomButton:

text:"Acceptar"

on_press: root.fdesenfocament (tabu.current_tab.text,median.text, gauss.text, radi.text, sigmac.text, sigma α .text,"a")

CustomButton:

text: "Cancel·lar"

on_press: root.fcancelar()

<umbral>:

imu:imu

title:"umbral"

name: "umbral_1"

GridLayout:

cols:1

size:root.width,root.height

size_hint:1,1

GridLayout:

cols:2

Image:

size_hint_x:0.55

source:"fotofinalr.jpg"

id:imu

GridLayout:

cols:1

size_hint_x:0.45

GridLayout:

cols:3

CustomLabel:

text:"Llindar"

Slider:

min:0

max:255

step: 1

id: umbralx

value:44

CustomLabel:

text:str(int(umbralx.value))

id:umbral

CustomLabel:

CustomLabel:

GridLayout:

cols:3

CustomButton:

text:"Previsualitzar"

```
on_press: root.fu (umbral.text,"p")
```

```
CustomButton:
```

```
text: "Acceptar"
```

```
on_press: root.fu (umbral.text, "a")
```

```
CustomButton:
```

```
text:"Cancel·lar"
```

```
on_press: root.fcancelar()
```

```
<erosio>:
```

```
title:"Erosió i Dilatació"
```

```
ime:ime
```

```
name:"erosio_1"
```

```
GridLayout:
```

```
cols:2
```

```
size:root.width,root.height
```

```
size_hint:1,1
```

```
Image:
```

```
id: ime
```

```
size_hint_x:0.55
```

```
source: "fotofinalr.jpg"
```

```
GridLayout:
```

```
cols:1
```

```
size_hint_x:0.45
```

```
GridLayout:
```

```
cols:2
```

```
CustomLabel:
```

```
text: "Rang matriu"
```

```
CustomSpinner:
```

```
text: "3"
```

```
values: ["3" , "5"]
```

```
id: erosios
```

```
GridLayout:
```

cols:3

CustomCheckBox:

id:sic1

CustomSpinner:

values: ["Erosió", "Dilatació"]

text:"Erosió"

id:sed1

CustomSpinner:

values: ["1","2","3","4"]

id:sv1

text: "4"

CustomCheckBox:

id:sic2

CustomSpinner:

values: ["Erosió", "Dilatació"]

id:sed2

text: "Erosió"

CustomSpinner:

values: ["1","2","3","4"]

id:sv2

text:"3"

CustomCheckBox:

id:sic3

CustomSpinner:

values: ["Erosió", "Dilatació"]

id:sed3

text: "Dilatació"

CustomSpinner:

values: ["1","2","3","4"]

id:sv3

text:"3"

CustomLabel:

GridLayout:

cols:3

size_hint_y: 0.5

CustomButton:

text: "Previsualització"

on_press: root.fed (erosios.text,
sic1.active,sed1.text,sv1.text,sic2.active,sed2.text,sv2.text,sic3.active,sed3.text,sv3.text, "p")

CustomButton:

text: "Acceptar"

on_press: root.fed (erosios.text,
sic1.active,sed1.text,sv1.text,sic2.active,sed2.text,sv2.text,sic3.active,sed3.text,sv3.text, "a")

CustomButton:

text: "Cancel·lar"

on_press: root.fcancelar ()

<enfoc>:

title:"Enfoc"

name:"enfoc_1"

imef:imef

GridLayout:

cols:2

size:root.width,root.height

size_hint:1,1

Image:

size_hint_x:0.55

source:"fotofinalr.jpg"

id:imef

GridLayout:

cols:1

size_hint_x:0.45

GridLayout:

cols:2

CustomLabel:

text:"tipus"

CustomSpinner:

text: "Enfocament"

values:["Enfocament","Pas alt"]

id:spv

GridLayout:

cols:1

CustomLabel:

GridLayout:

cols:3

CustomLabel:

CustomLabel:

CustomLabel:

CustomButton:

on_press: root.fenfoc(sp.v.text,"p")

text:"Previsualització"

CustomButton:

text:"Acceptar"

on_press: root.fenfoc(sp.v.text,"a")

CustomButton:

text:"Cancel·lar"

on_press: root.fcancelar()

<costats>:

title:"Costats"

name: "costats_1"

imc:imc

GridLayout:

cols:2

size:root.width,root.height

size_hint:1,1

Image:

id:imc

size_hint_x:0.55

source: "fotofinalr.jpg"

GridLayout:

cols:1

size_hint_x:0.45

GridLayout:

cols:1

TabbedPanel:

id:tabuc

do_default_tab: False

TabbedPanelItem:

text:"Canny"

GridLayout:

cols:3

size_hint_x:1

CustomLabel:

text: "Gradient \nmínim"

Slider:

min: 0

max: 255

id:gradmx

value:127

CustomLabel:

text:str(int(gradmx.value))

id:gradm

CustomLabel:

text: "Detecció \nGradient"

Slider:

min: 0

max: 255

id:graddx

value: 90

CustomLabel:

text:str(int(graddx.value))

id:gradd

TabbedPanelItem:

text:"Sobel"

GridLayout:

cols:1

GridLayout:

cols:2

CustomLabel:

text:"Rang matriu"

CustomSpinner:

values: ["3","5","7","9"]

text: "3"

id: sobelm

GridLayout:

cols:4

CustomLabel:

text: "X"

CheckBox:

id: cx

CustomLabel:

text:"Y"

CheckBox:

id: cy

TabbedPanelItem:

text: "Laplace"

GridLayout:

cols:2

CustomLabel:

text: "Rang matriu"

CustomSpinner:

text: "3"

values:["3", "5", "7", "9"]

id: laplaces

GridLayout:

cols:1

GridLayout:

cols:3

CustomLabel:

CustomLabel:

CustomLabel:

CustomButton:

text:"Previsualització"

on_press:

root.fcostats(tabuc.current_tab.text,gradm.text,gradd.text,sobel.m.text,cx.active,cy.active,laplaces.text, "p")

CustomButton:

text: "Acceptar"

```
        on_press:
root.fcstats(tabuc.current_tab.text,gradm.text,gradd.text,sobel.m.text,cx.active,cy.active,laplaces.te
xt, "a")
```

```
        CustomButton:
```

```
            on_press: root.fcancelar()
```

```
            text: "Cancel·lar"
```

```
<monedes>:
```

```
    title:"Monedes"
```

```
    imm:imm
```

```
    name: "monedes_1"
```

```
GridLayout:
```

```
    cols:2
```

```
    size:root.width,root.height
```

```
    size_hint:1,1
```

```
    Image:
```

```
        size_hint_x:0.55
```

```
        id:imm
```

```
        source: "fotofinalr.jpg"
```

```
GridLayout:
```

```
    cols:1
```

```
    size_hint_x:0.45
```

```
GridLayout:
```

```
    cols:1
```

```
    GridLayout:
```

```
        cols:2
```

```
GridLayout:
```

```
    cols:3
```

CustomLabel:

text: "Àrea"

Slider:

min:100

max:20000

id:areax

CustomLabel:

text:str(int(areax.value))

id: area

GridLayout:

cols:2

size_hint_y: 0.2

CustomLabel:

text:"moneda amb més area"

CustomSpinner:

text:"2EUR"

values:["2EUR","50ct","1EUR","20ct","5ct","10ct","2ct","1ct"]

id:moneda

GridLayout:

cols:2

CustomLabel:

text:"2 euros"

CustomLabel:

text:""

id:dos

CustomLabel:

text: "1 euro"

CustomLabel:

text: ""

id:u

CustomLabel:

text: "50 cèntims"

CustomLabel:

text:""

id:cinquanta

CustomLabel:

text: "20 cèntims"

CustomLabel:

text:""

id:vint

CustomLabel:

text: "10 cèntims"

CustomLabel:

text:""

id:deu

CustomLabel:

text: "5 cèntims"

CustomLabel:

text:""

id:cinc

CustomLabel:

text: "2 cèntims"

CustomLabel:

text:""

id:dosc

CustomLabel:

text: "1 cèntim"

CustomLabel:

text:""

id:uc

GridLayout:

cols:3

CustomLabel:

text: "suma total:"

CustomLabel:

text:""

id:sumatotal

CustomLabel:

text:"€"

GridLayout:

cols:1

CustomLabel:

text:""

text_size:root.width*0.28,None

id:dades

CustomLabel:

text:""

text_size:root.width*0.28,None

id:errorl

GridLayout:

cols:3

CustomButton:

text:"Paràmetres\nestandards"

font_size:"15dp"

on_press:

root.fmonedes(area.text,"p",dos.text,u.text,cinquanta.text,sumatotal.text,vint.text,deu.text,cinc.text,dosc.text,uc.text, moneda.text, dates.text,errorl.text)

CustomButton:

text: "Paràmetres\n establerts\n per l'usuari"

```
on_press: root.fmonedes(area.text,  
"a", dos.text, u.text, cinquanta.text, sumatotal.text, vint.text, deu.text, cinc.text, dosc.text, uc.text,  
moneda.text, dades.text, errorl.text)
```

```
font_size: "15dp"
```

CustomButton:

```
on_press: root.fcancelar()
```

```
text: "Cancel·lar"
```

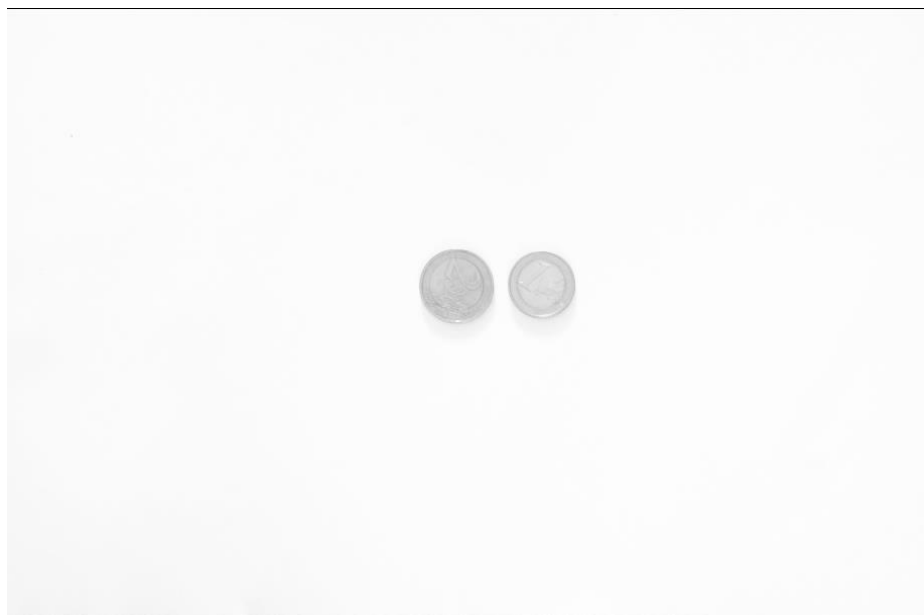
```
font_size: "15dp"
```

5 ANNEX 4: Proves de l'aplicació

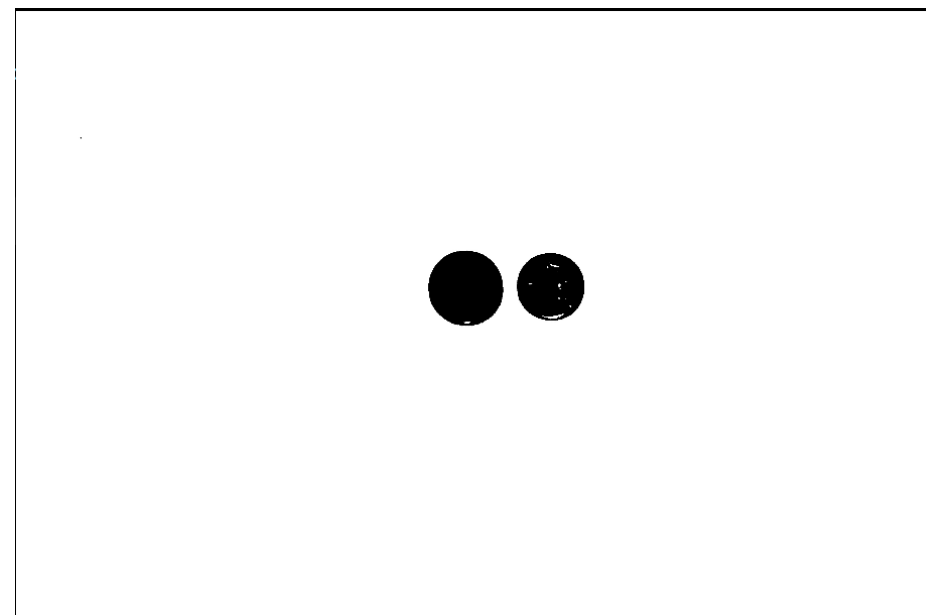
DIFERENTS FONTS



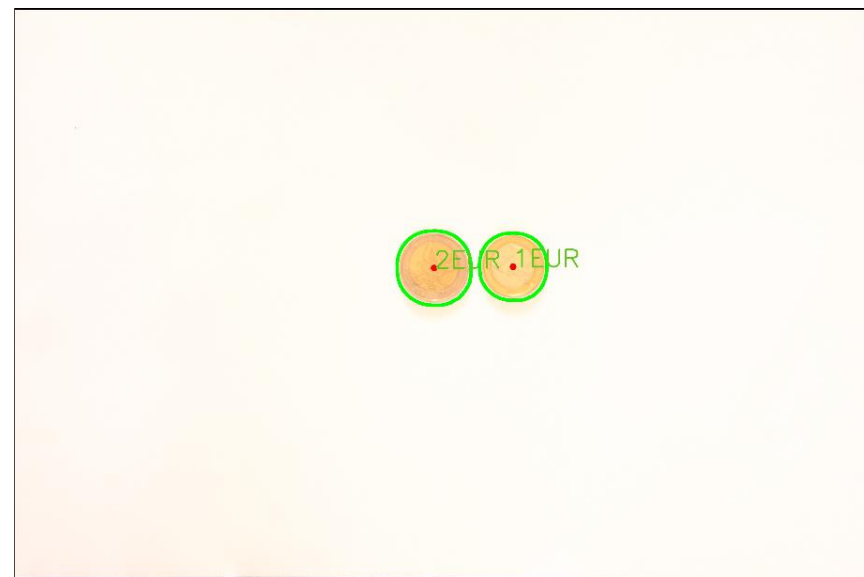
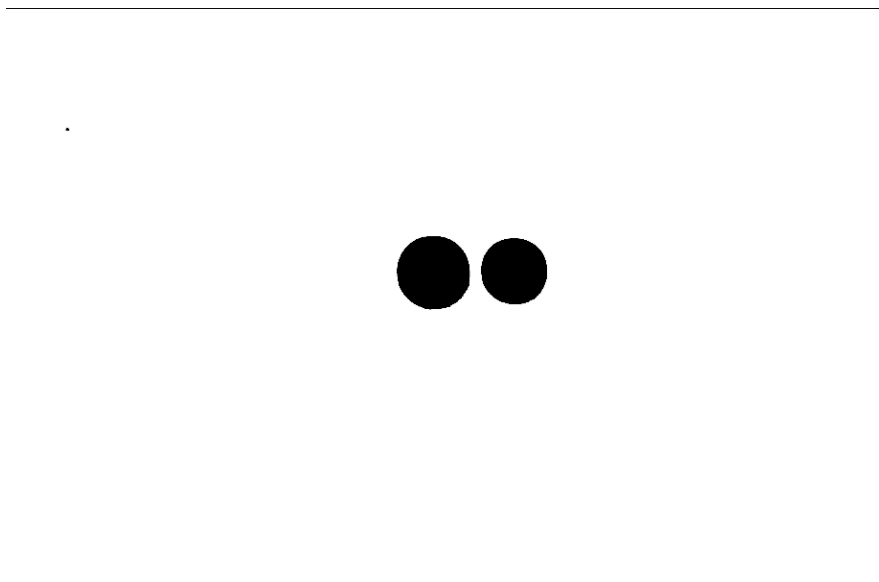
Experiment	Fons Blanc EV0, 2 Euros, 1 Euro,	Codi experiment	FBEV0_R0	Captura	Fujifilm Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7, EV 0
Condicions processat	EV0, filtre blanc i negre, lllindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	No detecta les monedes amb els paràmetres estàndards.



Filtre blanc i negre

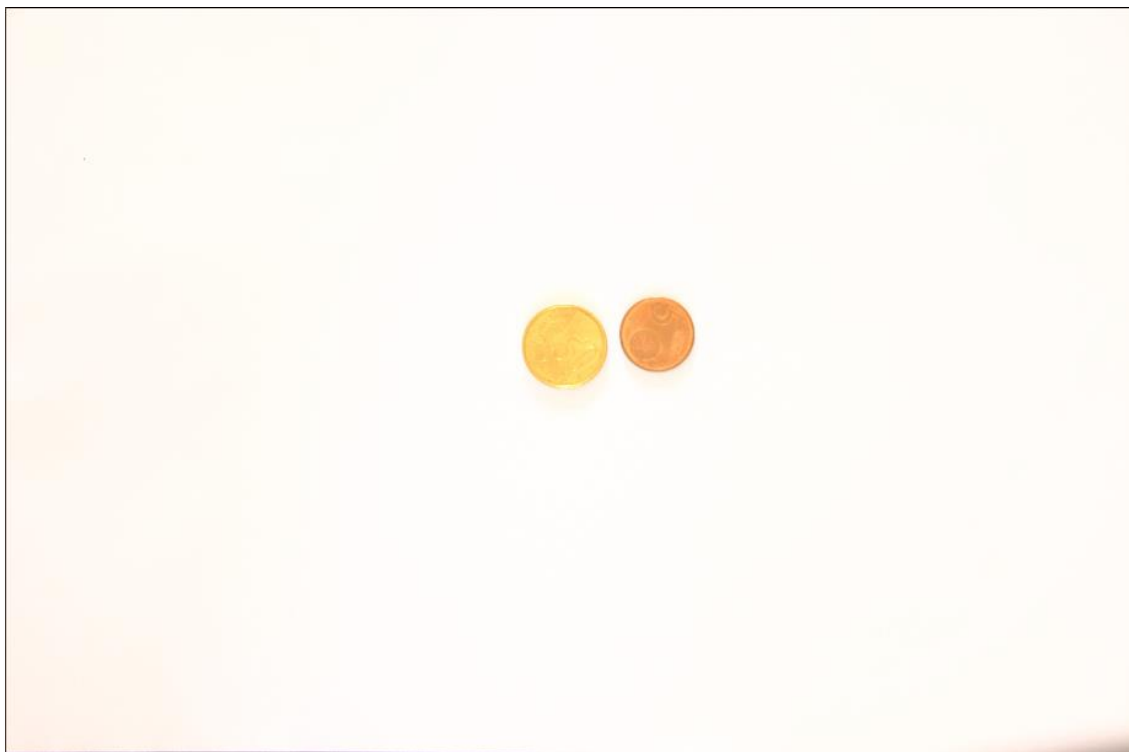


Umbral

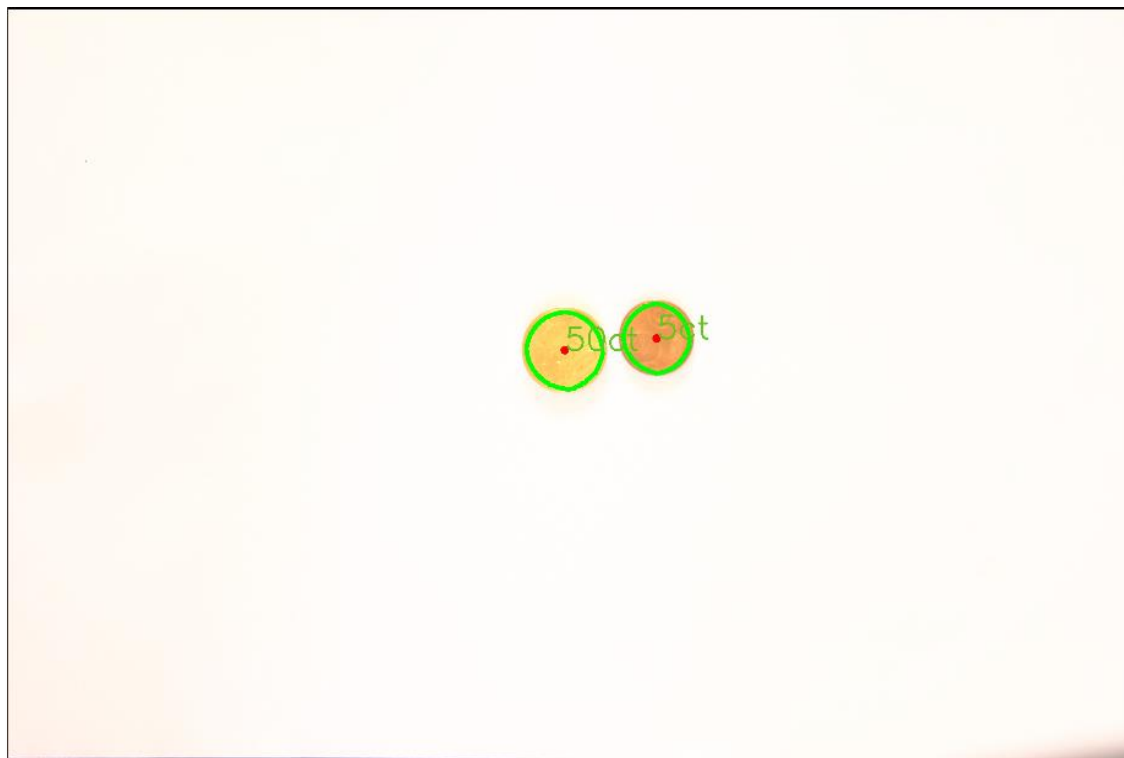


Erosió i dilatació

Experiment	Fons Blanc EV0, 2 Euros, 1 Euro	Codi experiment	FBEV0_R1	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7, EV 0
Condicions processat	EV 0, filtre blanc i negre, llindar 226, erosió 4, erosió 3, erosió 3, àrea mínima 8387.			Resultats	La moneda al ser metàl·lica reflexa la llum i té uns valors de color molt elevats, propers al blanc. Per aquest motiu el valor del llindar s'ha hagut de pujar per poder detectar la seva superfície. Les monedes es detecten amb els paràmetres establerts manualment.



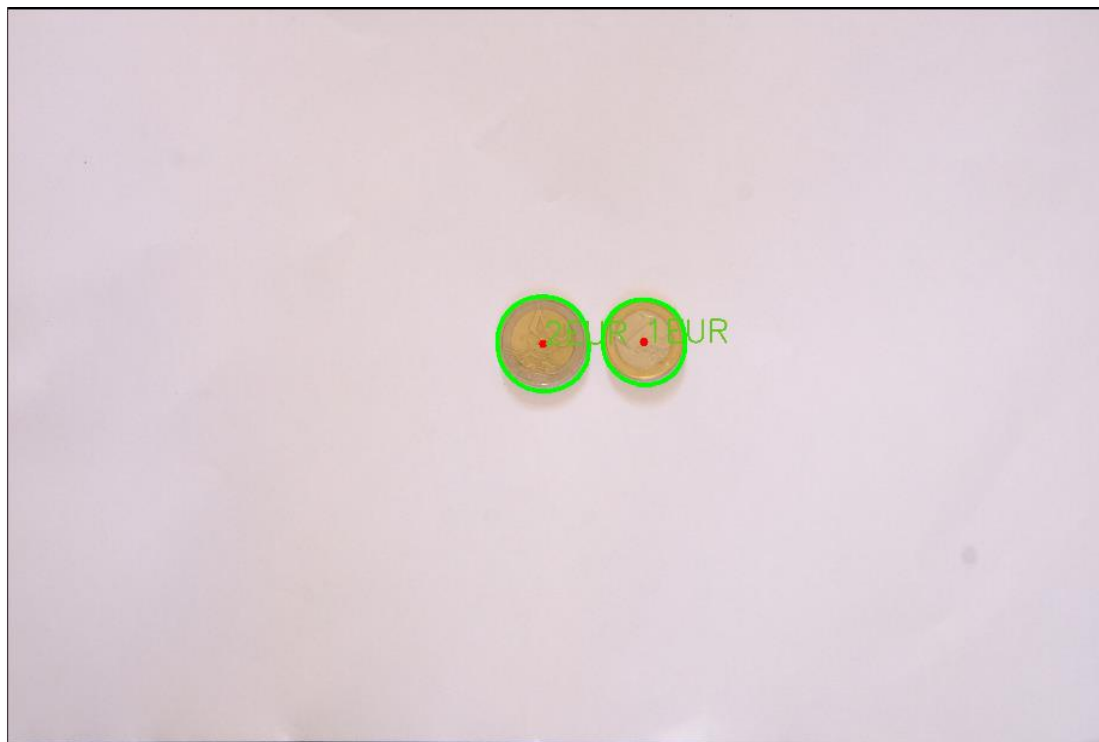
Experiment	Fons Blanc EVO, 50 cèntims, 5 cèntims,	Codi experiment	FB2EVO_R0	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7, EV 0
Condicions processat	EVO, filtre blanc i negre, lllindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	No detecta les monedes amb els paràmetres estàndards.



Experiment	Fons Blanc EVO, 50 cèntims, 5 cèntims	Codi experiment	FB2EVO_R1	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7, EV 0
Condicions processat	EVO, filtre blanc i negre, llindar 242, erosió 4, erosió 1, dilatació 3, dilatació4, dilatació 3, dilatació 4, dilatació 3, dilatació 4, àrea mínima 100.			Resultats	Les monedes es de 50 cèntims i 5 cèntims reflecteixen més la llum ja que el valor del llindar és mes gran respecte l'experiment FBEVO_R1. Les monedes es detecten amb els paràmetres establerts manualment.



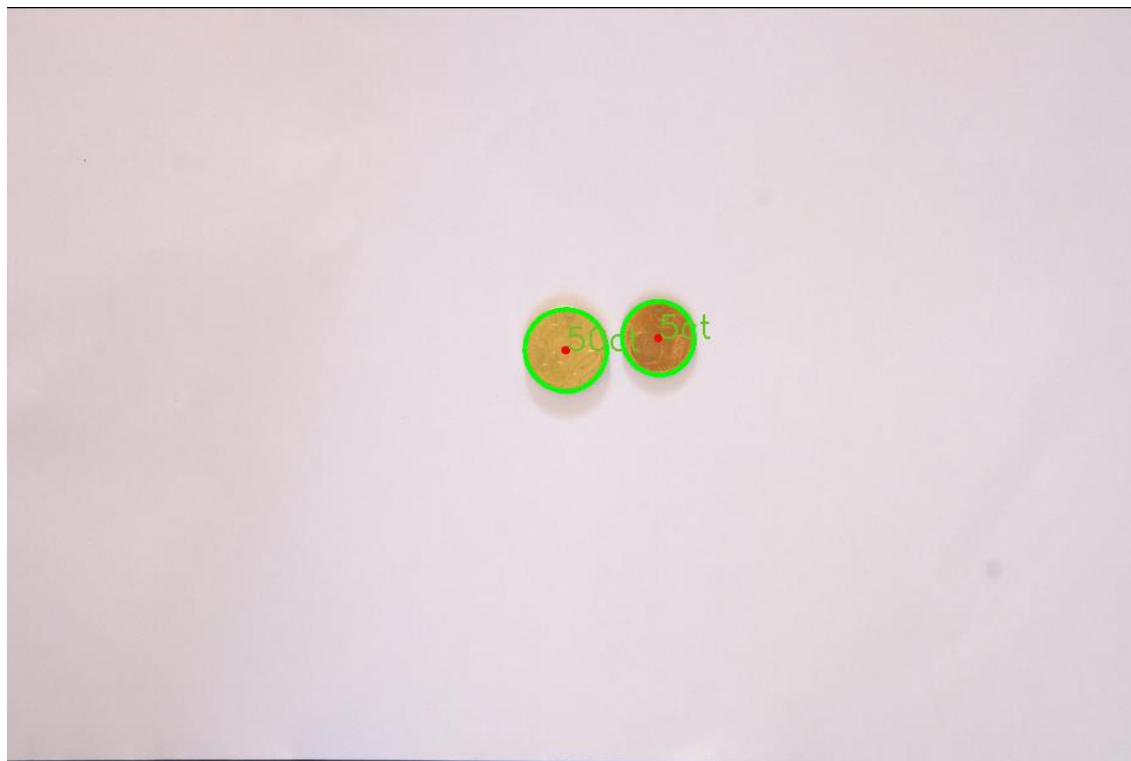
Experiment	Fons Blanc EV-1, 2 Euros, 1 Euro	Codi experiment	FBEV-1_R0	Captura	Fujifilm Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7, EV - 1
Condicions processat	EV-1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	No detecta les monedes amb els paràmetres estàndards.



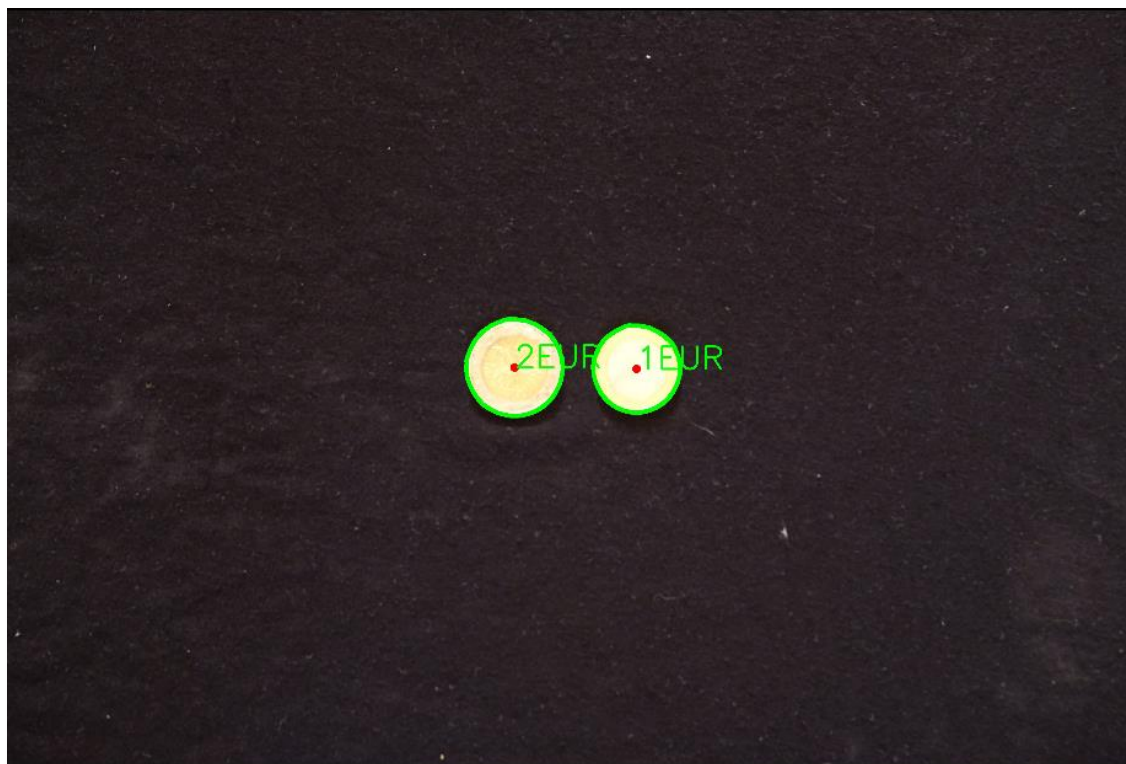
Experiment	Fons Blanc EV-1, 2 Euros, 1 Euro	Codi experiment	FBEV-1_R1	Captura	Fujifilm Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7, EV - 1
Condicions processat	EV-1, filtre blanc i negre, llindar 182, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 2, dilatació 3, àrea mínima 100.			Resultats	Com que la quantitat de llum és més petita el valor del llindar també és més petit respecte experiment FBEV0_R1. Les monedes es detecten amb els paràmetres establerts manualment.



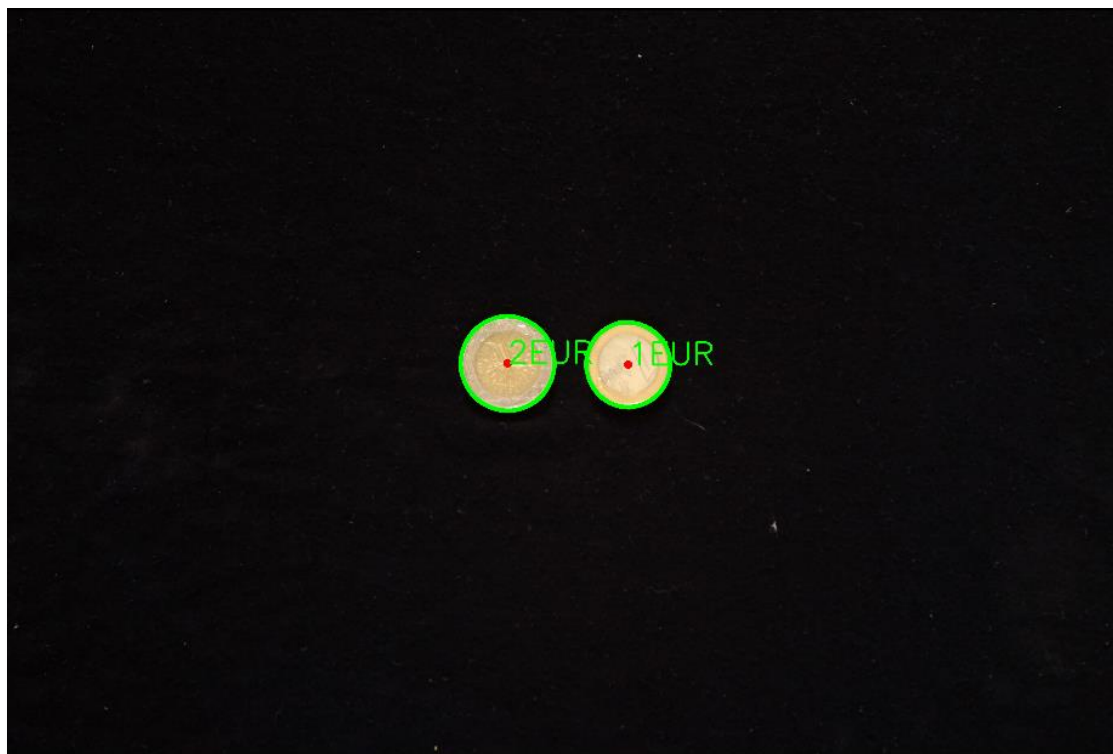
Experiment	Fons Blanc EV-1, 50 cèntims, 5 cèntims,	Codi experiment	FB2EV-1_R0	Captura	Fujifilm Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7, EV - 1
Condicions processat	EV-1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	No detecta les monedes amb els paràmetres estàndards.



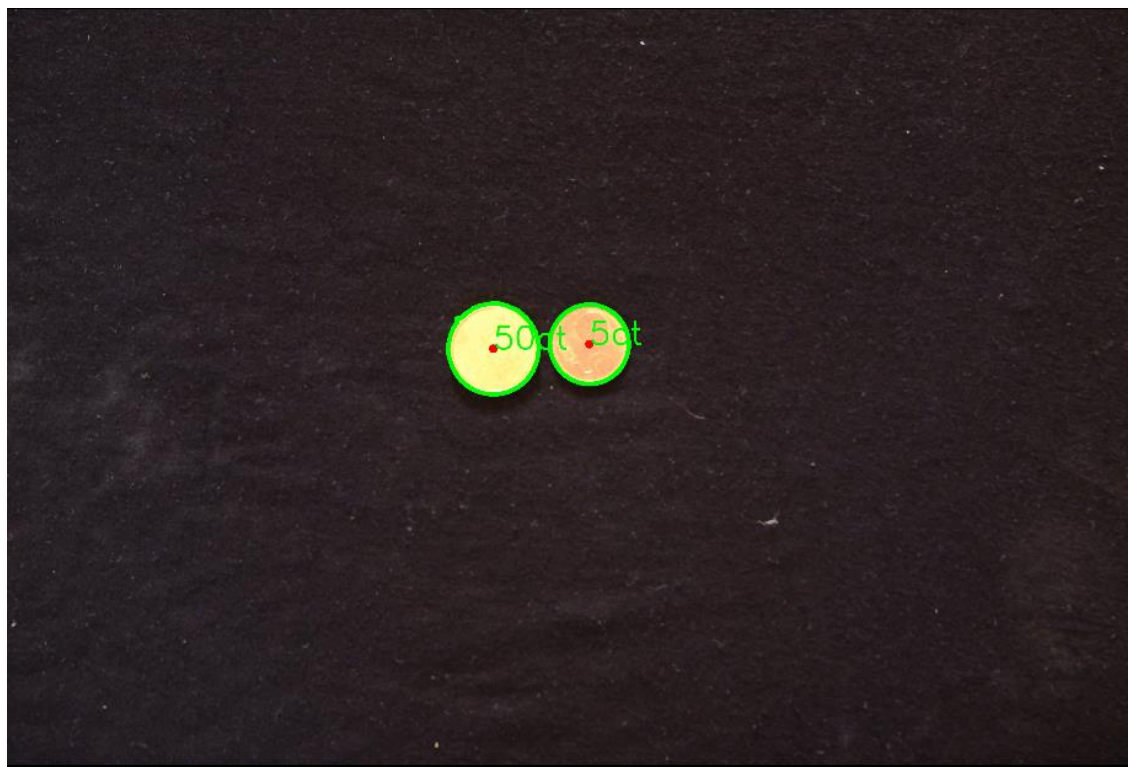
Experiment	Fons Blanc EV-1, 50 cèntims, 5 cèntims	Codi experiment	FB2EV-1_R1	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7, EV - 1
Condicions processat	EV-1, filtre blanc i negre, llindar 180, erosió 4, erosió 1, dilatació 3, erosió 4, dilatació 3, dilatació 4, erosió 4, erosió 3, dilatació 3, dilatació 4, dilatació 3, àrea mínima 100.			Resultats	Com que la quantitat de llum és més petita el valor del llindar també es mes petit, respecte experiment FB2EV-1_R1. Les monedes es detecten amb els paràmetres establerts manualment.



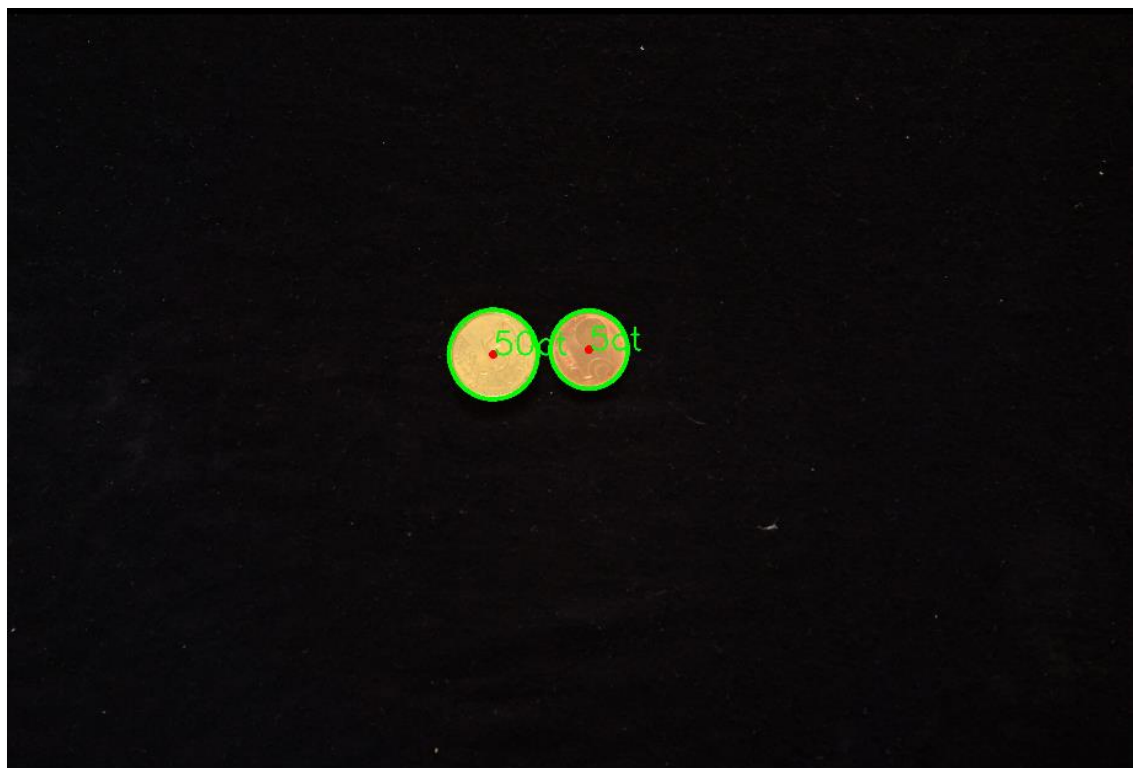
Experiment	Fons negre EV0,2 Euros, 1 Euro	Codi experiment	FNEVO_R0	Captura	Fujifilm Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV+1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres estàndards.



Experiment	Fons negre EV-1, 2 Euros, 1 Euro,	Codi experiment	FNEV-1_R0	Captura	Fujifilm Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV+1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres estàndards.



Experiment	Fons negre EVO, 50 cèntims, 5 cèntims	Codi experiment	FN2EVO_R0	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EVO, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres estàndards.



Experiment	Fons negre EV-1, 50 cèntims, 5 cèntims	Codi experiment	FN2EV-1_R0	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV-1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres estàndards.



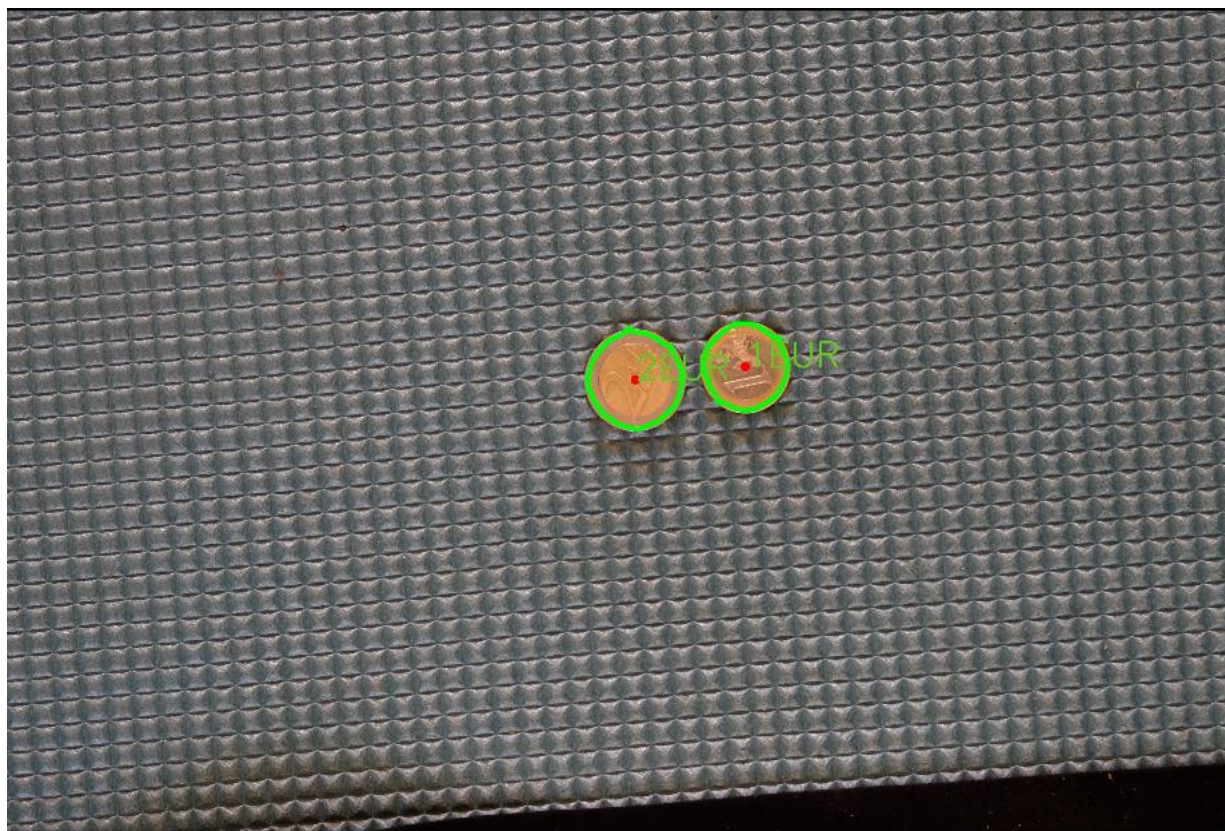
Experiment	Fons Relleu EVO, 2 Euros, 1 Euro.	Codi experiment	FREVO_R0	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distancia focal 63.8 EV 0
Condicions processat	EVO, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	No detecta les monedes amb els paràmetres estàndards.



Experiment	Fons relleu, EVO, 2 Euros, 1 Euro.	Codi experiment	FREVO_R1	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 63,8, EV 0
Condicions processat	EV+1, filtre blanc i negre, llindar 182, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 2, dilatació 3, àrea mínima 100.			Resultats	Les monedes es detecten amb els paràmetres establerts manualment.



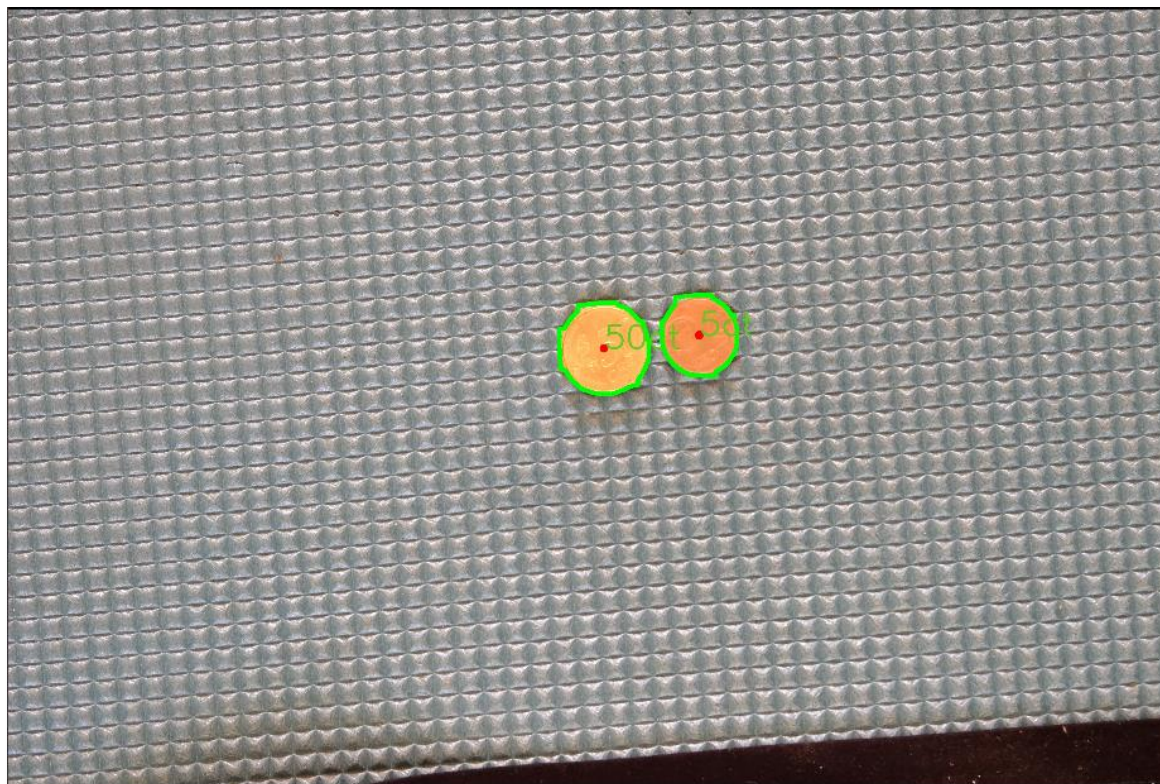
Experiment	Fons relleu, EV -1, 2 Euros, 1 Euro,	Codi experiment	FREV-1_R0	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 63,8 EV -1
Condicions processat	EV-1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	No detecta les monedes amb els paràmetres estàndards.



Experiment	Fons Relleu EV-1, 2 Euros, 1 Euro	Codi experiment	FREV-1_R1	Captura	Fujifilm Xt-20, ISO 2500, Velocitat: 1/8, Distancia focal 63,8, EV - 1
Condicions processat	EV-1, filtre blanc i negre, llindar 127, desenfocament promig 11 , erosió 4, erosió 3, erosió 3, àrea mínima 100.			Resultats	Les monedes es detecten amb els paràmetres establerts manualment.



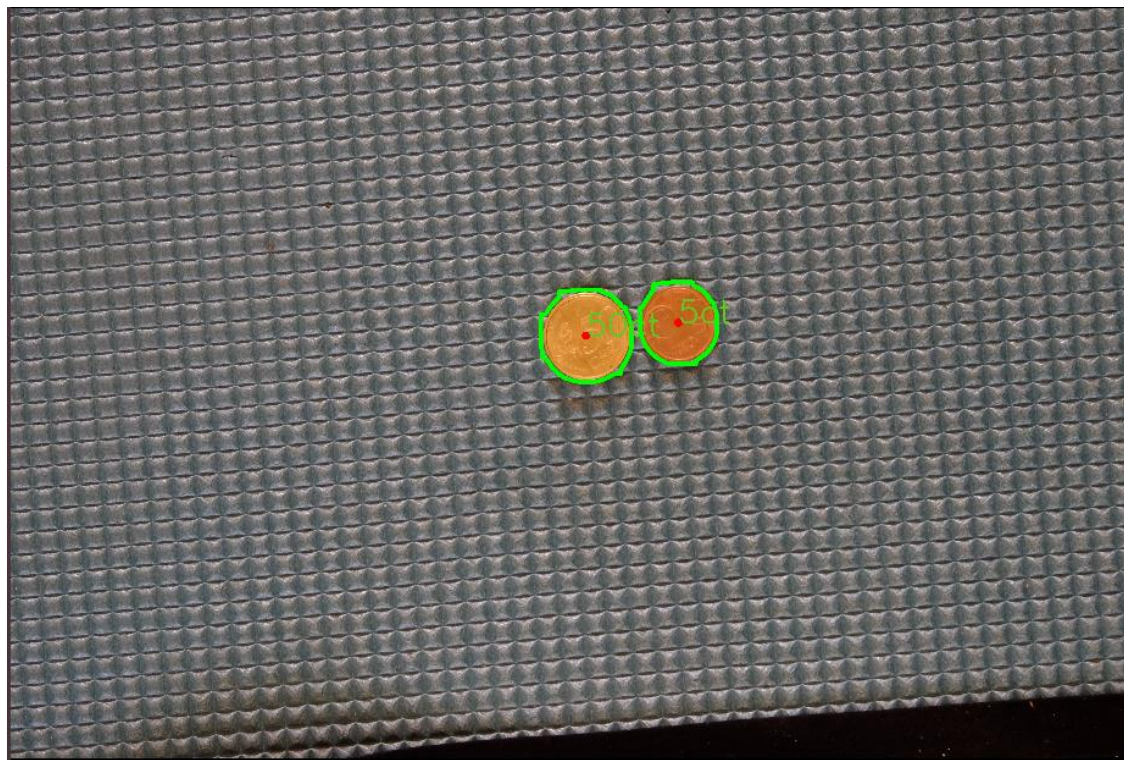
Experiment	Fons Relleu, EV0, 50 cèntims, 5 cèntims	Codi experiment	FR2EV0_R0	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 63.8, EV 0
Condicions processat	EVO, filtre blanc i negre, lllindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	No detecta les monedes amb els paràmetres estàndards.



Experiament	Fons Relleu, EV0, 50 cèntims, 5 cèntims	Codi experiment	FR2EV0_R1	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 63.8, EV 0
Condicions processat	EV0, filtre blanc i negre, desenfocament promig 11, llindar 148, erosió 4, erosió 3, erosió 4, dilatació 4, dilatació 3, dilatació 4, àrea mínima 100.			Resultats	Les monedes es detecten amb els paràmetres establerts manualment.

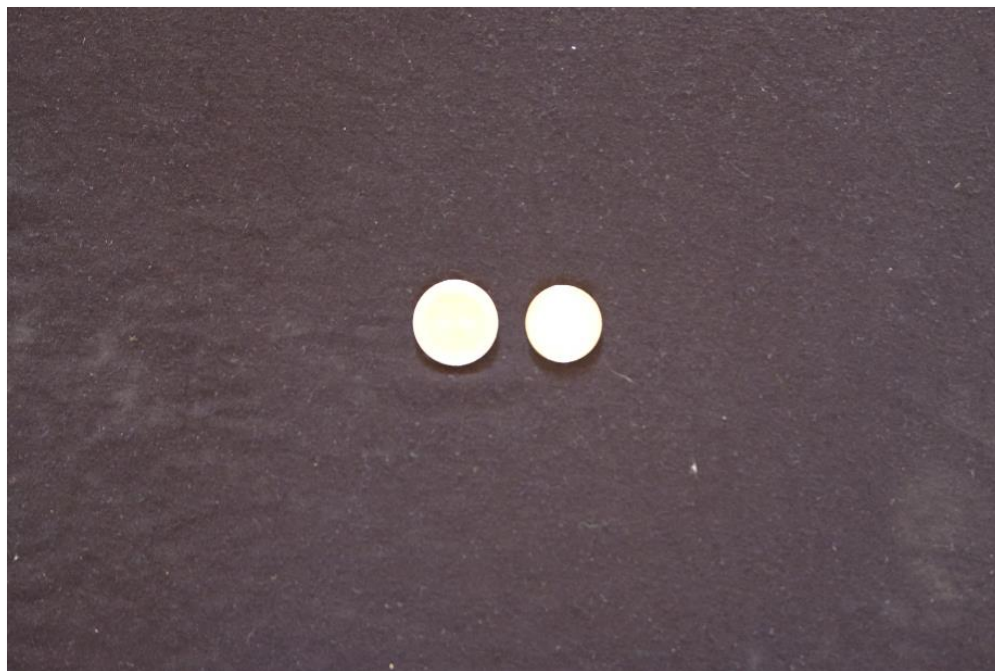


Experiment	Fons Relleu, EV-1, 50 cèntims, 5 cèntims	Codi experiment	FR2EV-1_R0	Captura	Fujifilm Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 63.8, EV - 1
Condicions processat	EV-1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	No detecta les monedes amb els paràmetres estàndards.

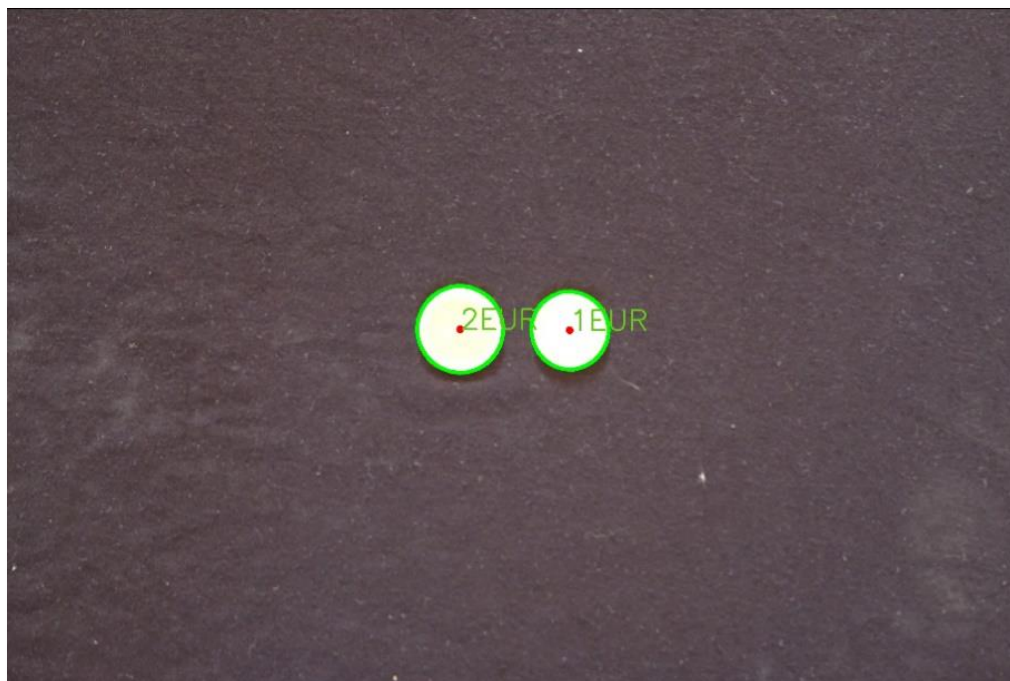


Experiment	50 cèntims, 5 cèntims, Fons Rellu, EV-1	Codi experiment	FR2EV-1_R1	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 63.8 EV -1
Condicions processat	EV-1, filtre blanc i negre, desenfocament promig 11, lllindar 108, erosió 4, erosió 3, erosió 4, dilatació 4, erosió 3, dilatació 4, erosió 3, dilatació 3, erosió 4, erosió 4, dilatació 3, àrea mínima 2420.			Resultats	Les monedes es detecten amb els paràmetres establerts manualment.

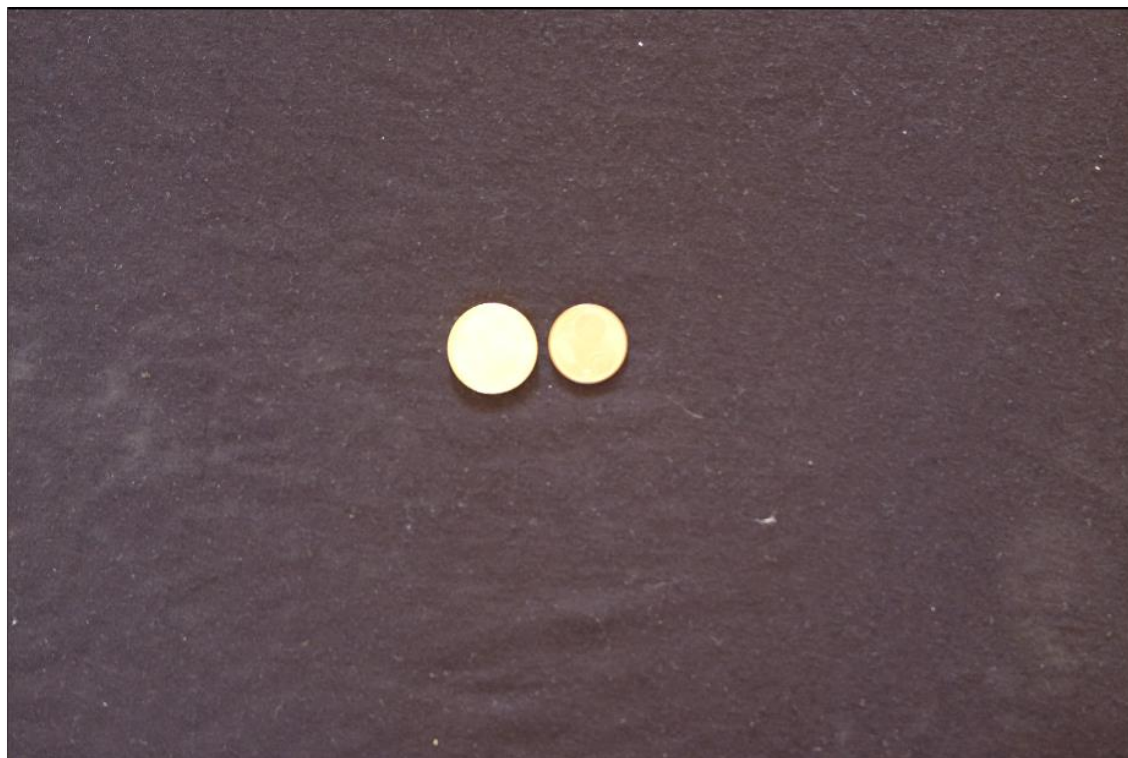
QUANTITAT DE LLUM



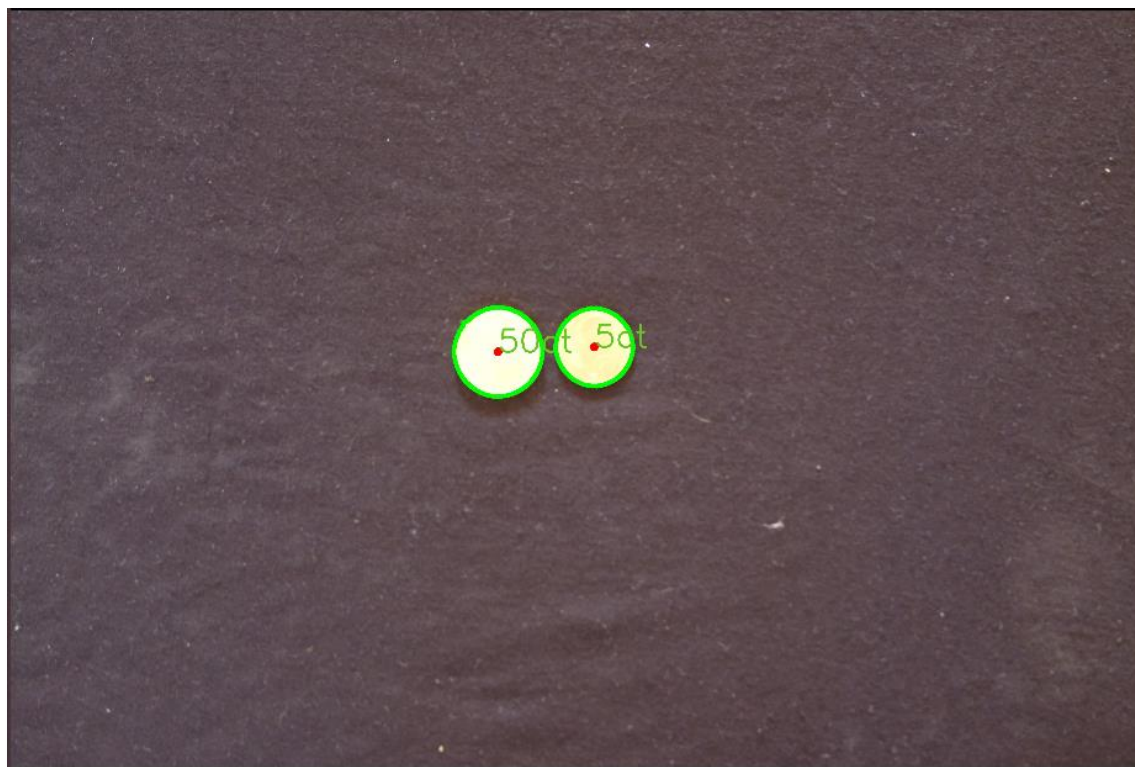
Experiment	Quantitat llum EV+1, 2 Euros, 1 Euro, EV+1	Codi experiment	EV1_R0	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV+1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	No detecta les monedes amb els paràmetres estàndards.



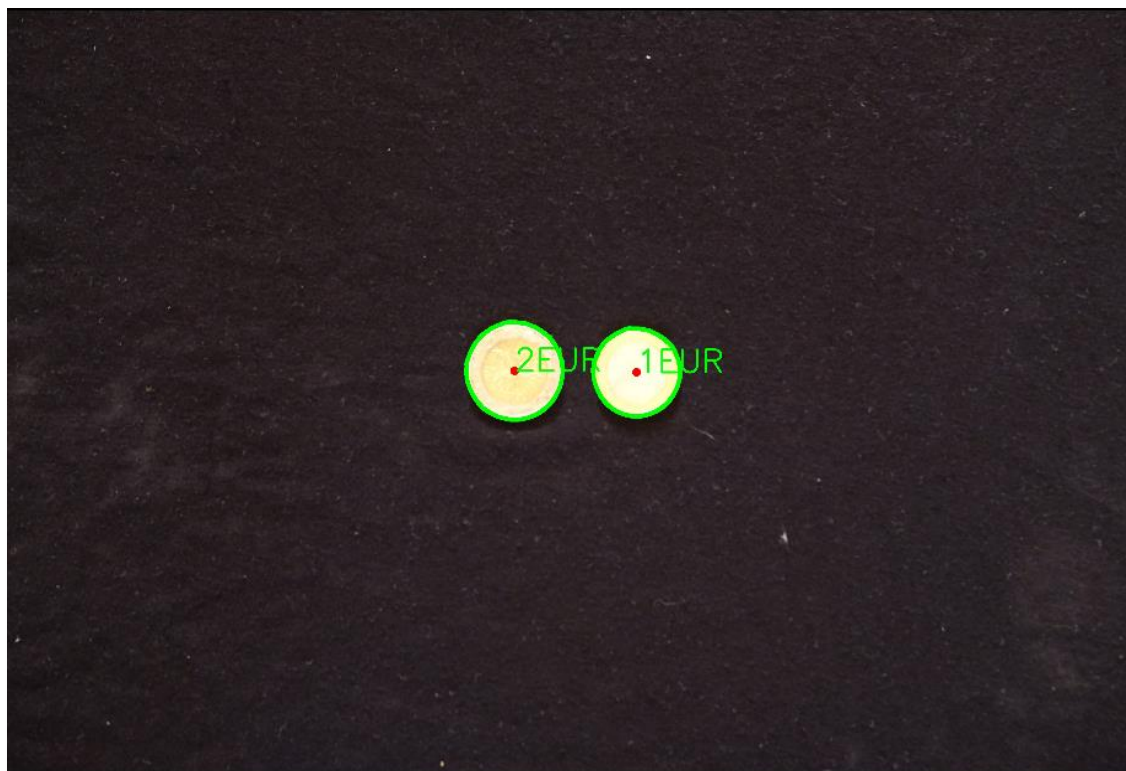
Experiment	Quantitat llum EV+1, 2 Euros, 1 Euro	Codi experiment	EV1_R1	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV+1, filtre blanc i negre, llindar 11, erosió 4, erosió 2, dilatació 3, àrea mínima: 8400.			Resultats	Les monedes es detecten amb els paràmetres establerts manualment.



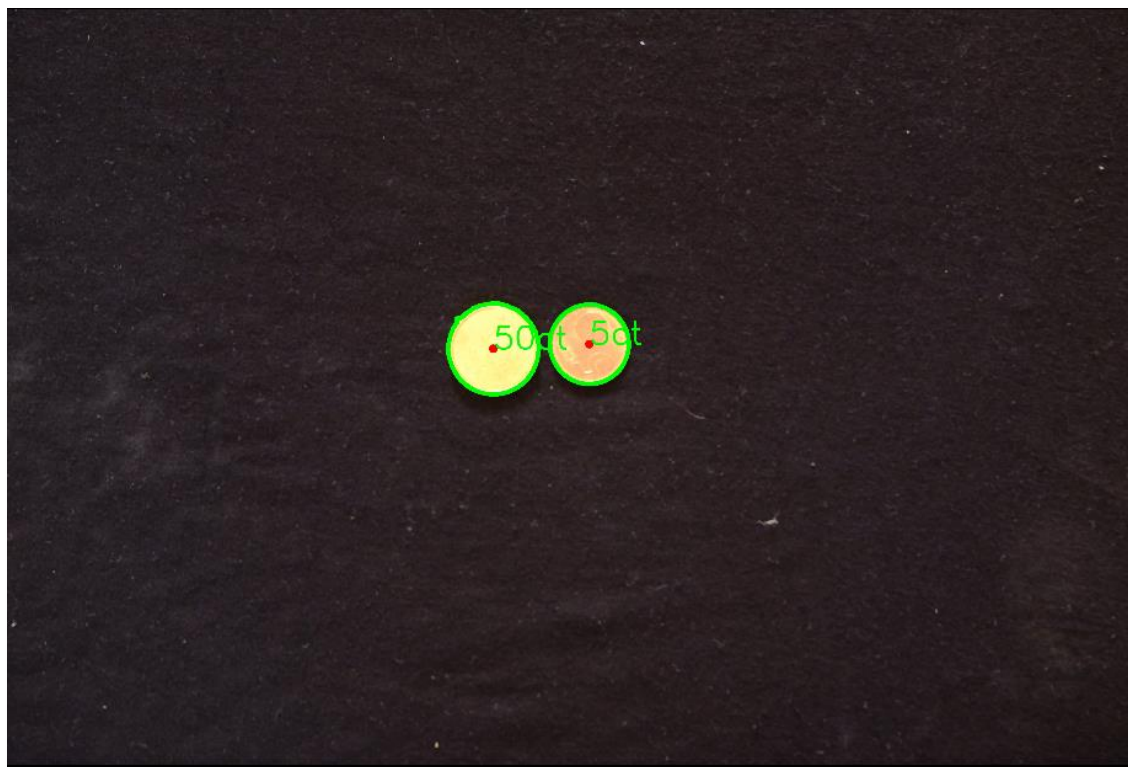
Experiment	Quantitat llum EV+1, 50 cèntims, 5 cèntims	Codi experiment	EV1(2)_R0	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV+1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	No detecta les monedes amb els paràmetres estàndards.



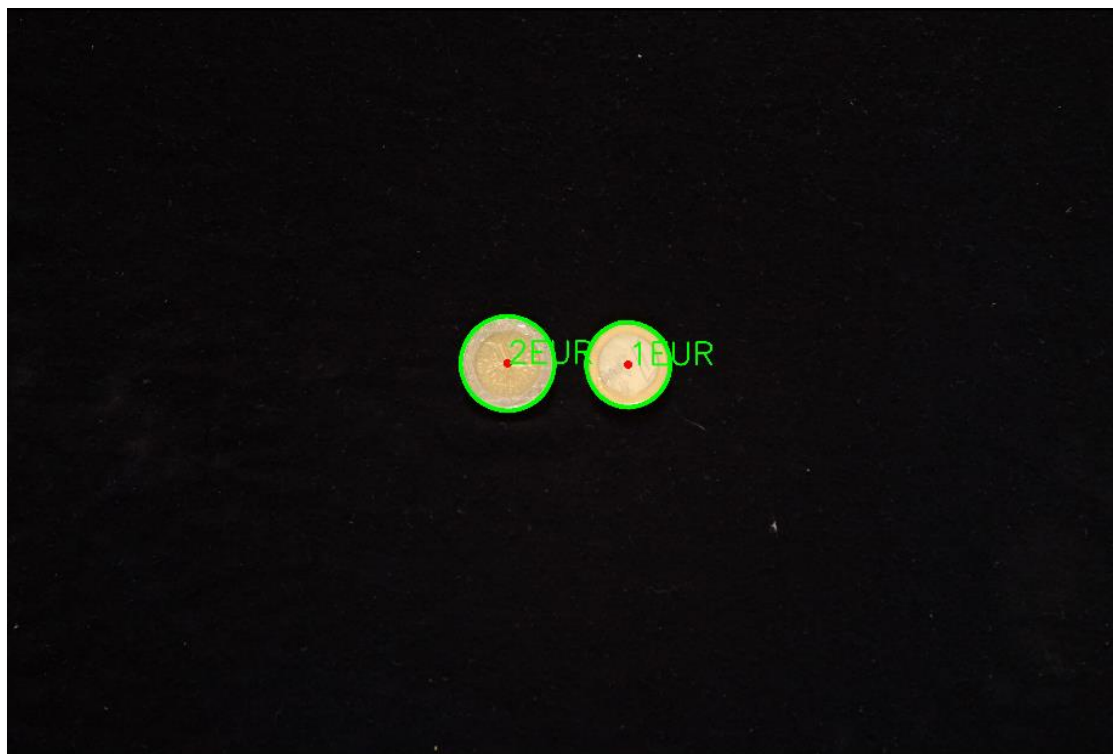
Experiment	Quantitat llum EV+1, 50 cèntims, 5 cèntims	Codi experiment	EV1(2)_R1	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV+1, filtre blanc i negre, llindar 105, erosió 3, erosió 4, dilatació 4, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres establerts per l'usuari.



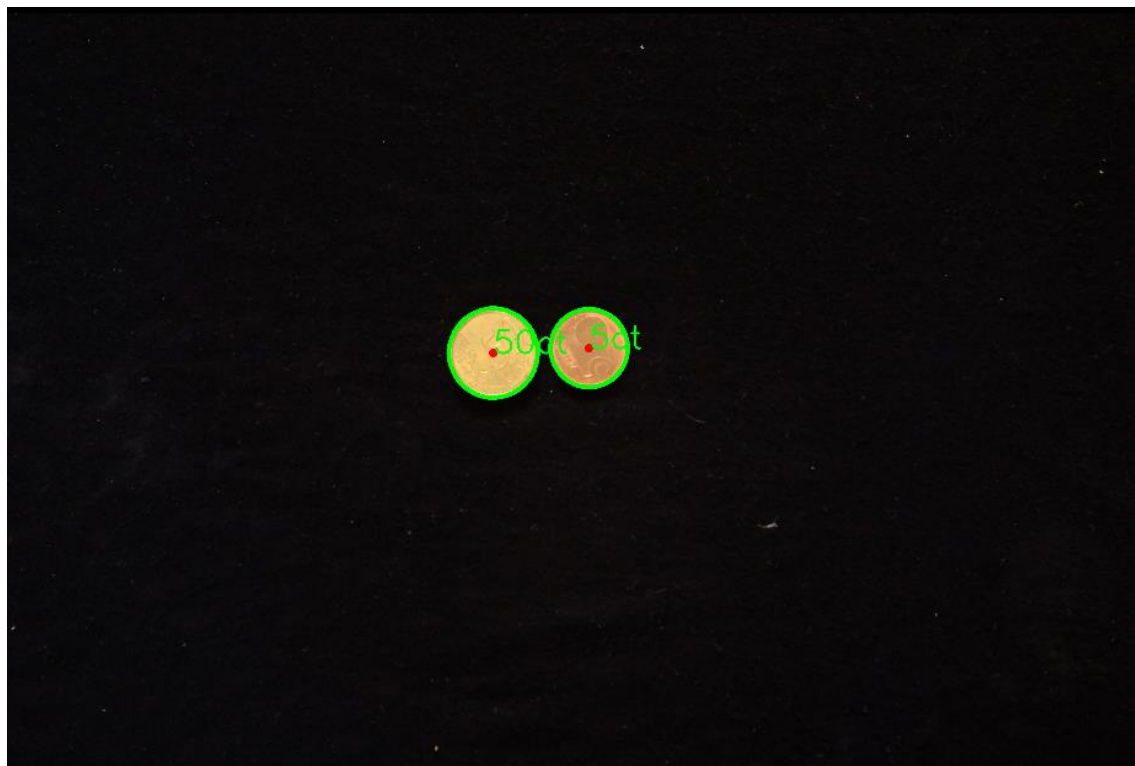
Experiment	Quantitat llum EVO, 2 Euros, 1 Euro	Codi experiment	EVO_R0	Captura	Fujifilm Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV+1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres estàndards.



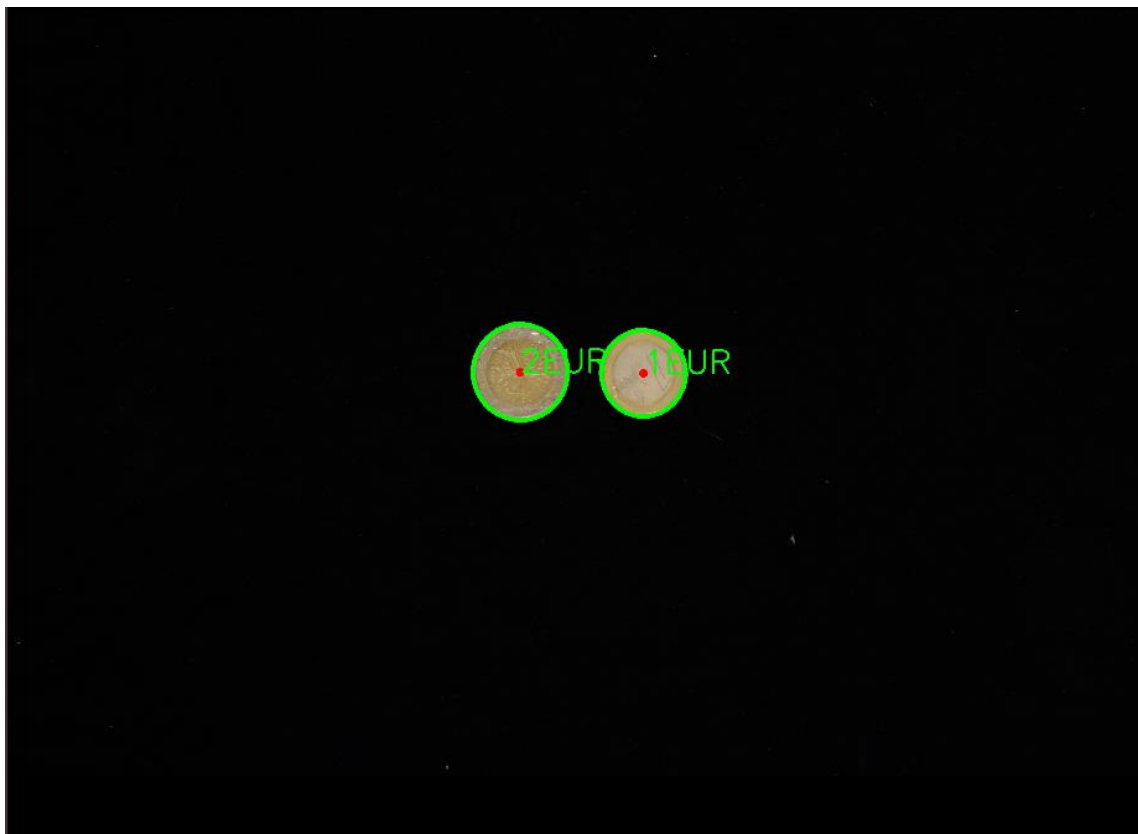
Experiment	Quantitat llum EVO, 50 cèntims, 5 cèntims	Codi experiment	EVO(2)_R0	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EVO , filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres estàndards.



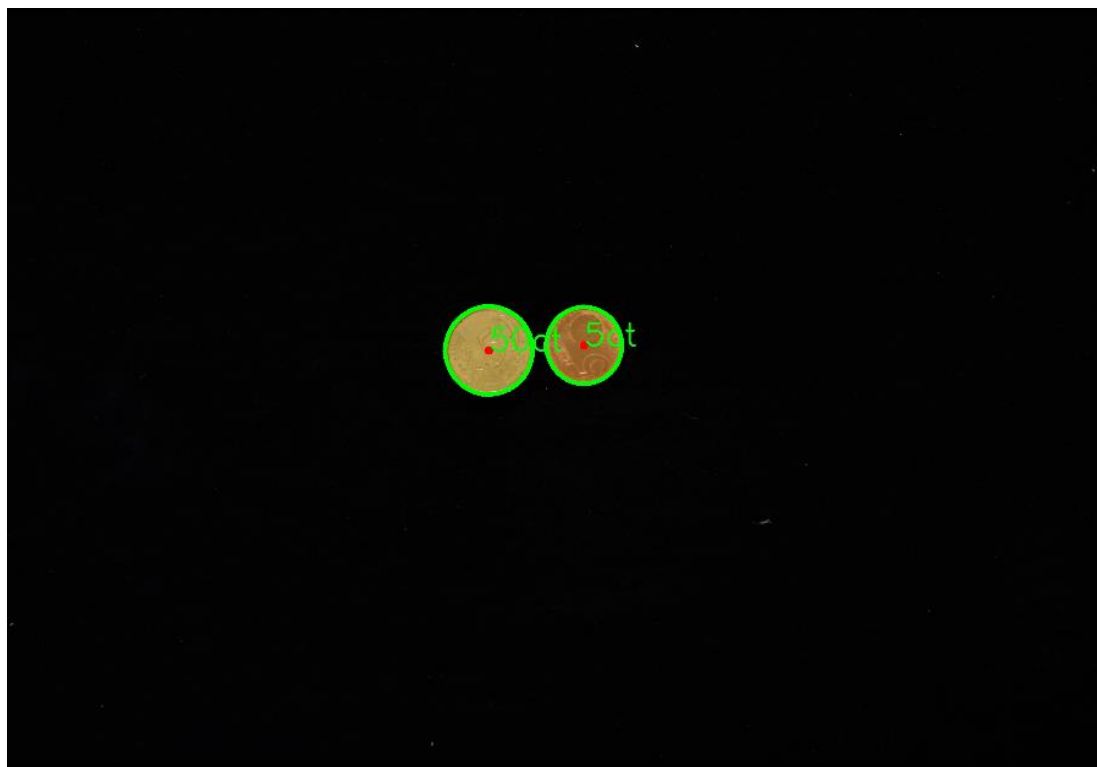
Experiment	Quantitat llum EV-1, 2 Euros, 1 Euro	Codi experiment	EV-1_R0	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV+1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres estàndards.



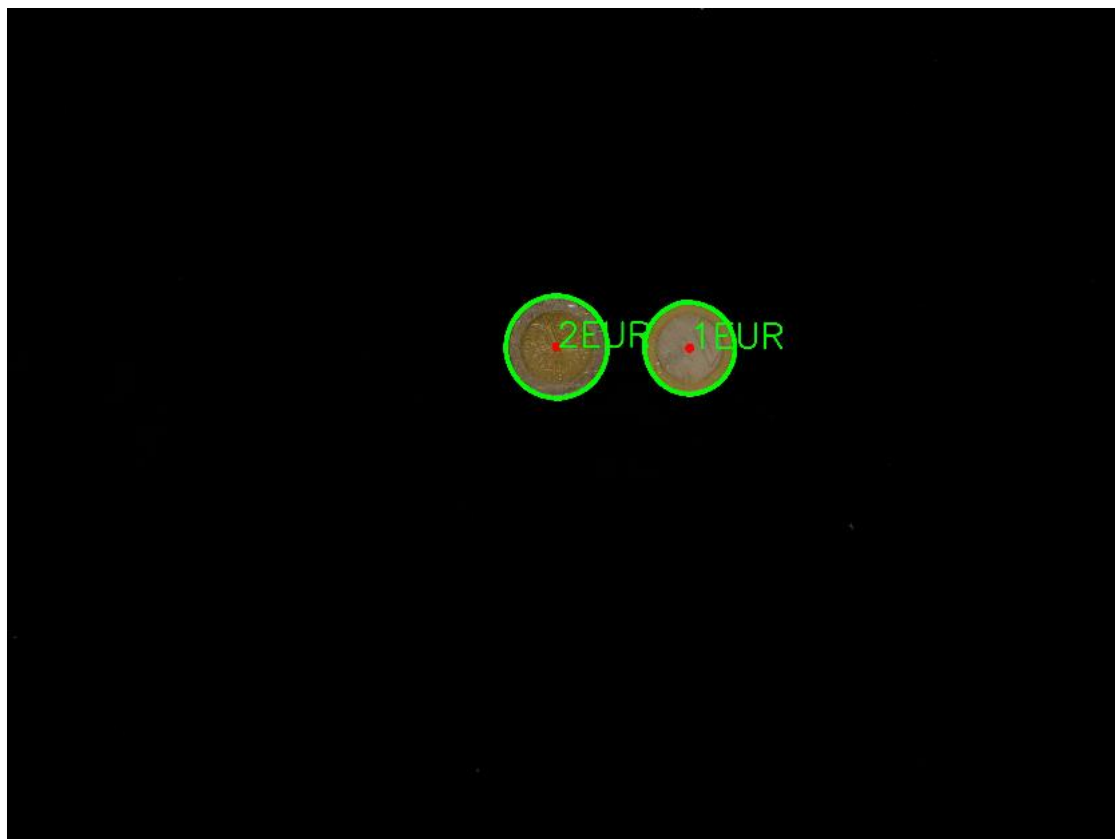
Experiment	Quantitat llum EV-1, 50 cèntims, 5 cèntims	Codi experiment	EV-1(2)_R0	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV-1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres estàndards.



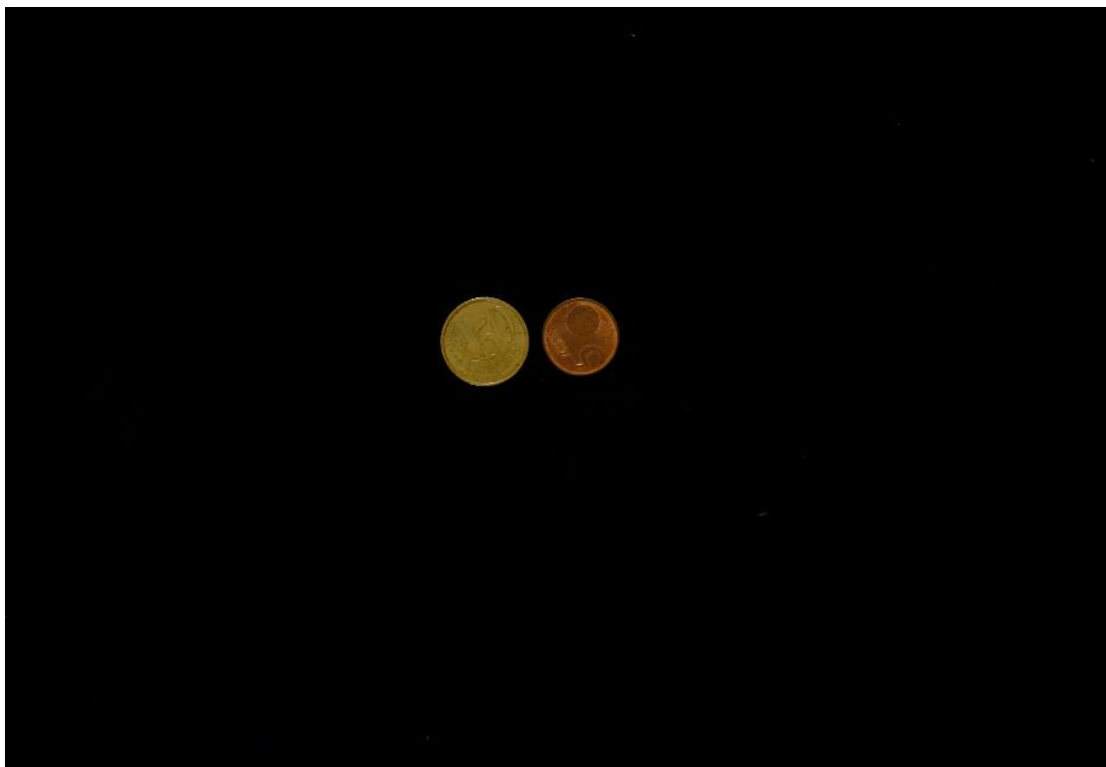
Experiment	Quantitat llum EV-2, 2 Euros, 1 Euro	Codi experiment	EV-2_R0	Captura	Fujifilm Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV+1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres estàndards.



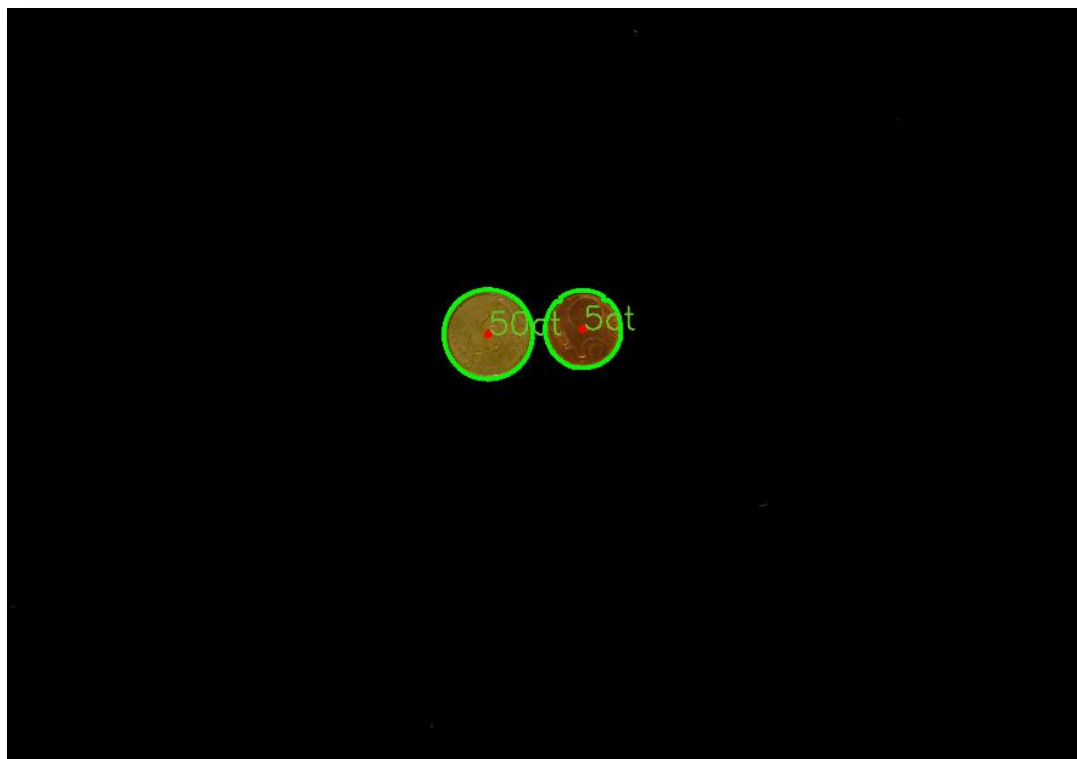
Experiment	Quantitat llum EV-2, 50 cèntims, 5 cèntims	Codi experiment	EV-2(2)_R0	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV-2, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres estàndards.



Experiment	Quantitat llum EV-3, 2 Euros, 1 Euro	Codi experiment	EV-3_R0	Captura	Fujifilm Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV+1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres estàndards.

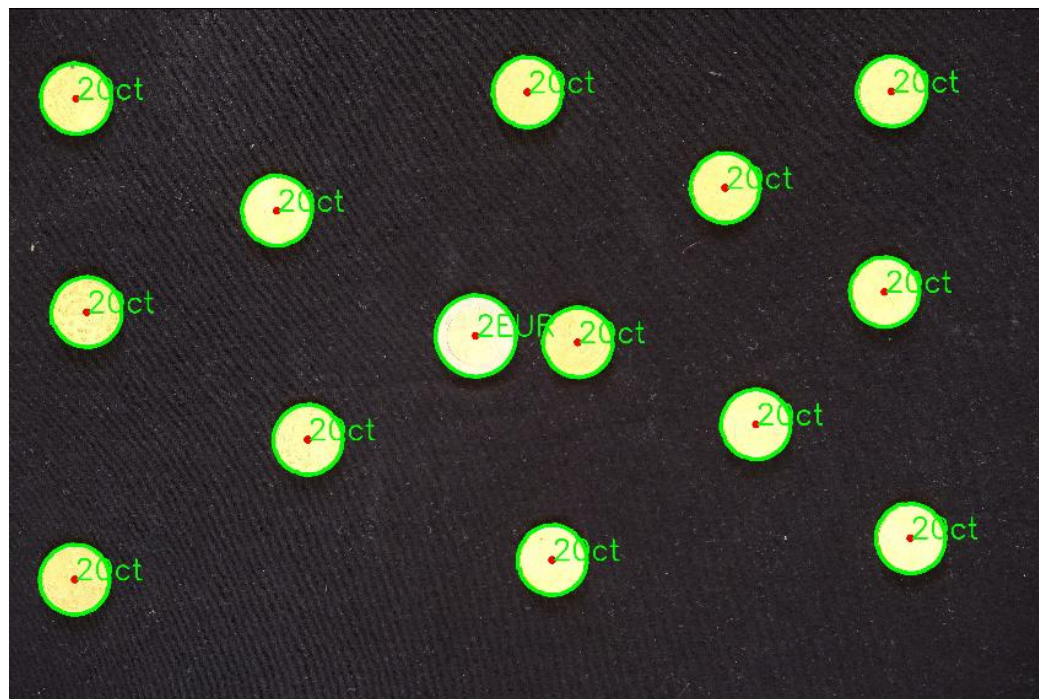


Experiment	Quantitat llum EV-3, 50 cèntims, 5 cèntims	Codi experiment	EV-3(2)_R0	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV-3, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	No detecta les monedes amb els paràmetres estàndards.

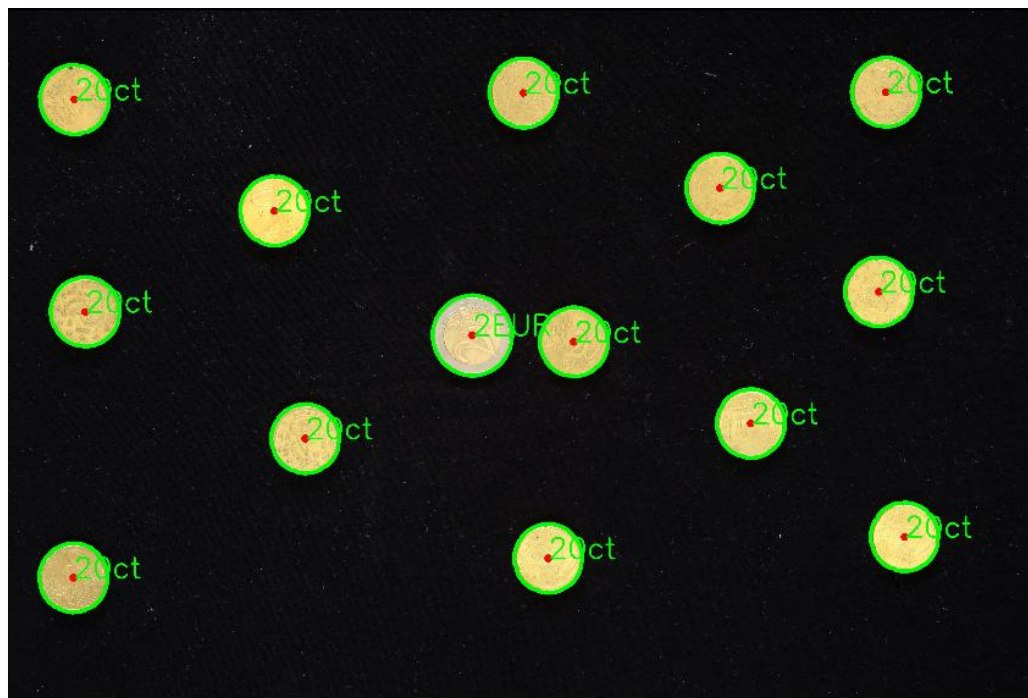


Experiment	Quantitat llum EV-3, 50 cèntims, 5 cèntims	Codi experiment	EV-3(2)_R1	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7
Condicions processat	EV-3, filtre blanc i negre, llindar 50, dilatació 1, erosió 4, dilatació 4, àrea mínima 100.			Resultats	Les monedes es detecten amb els paràmetres establerts manualment.

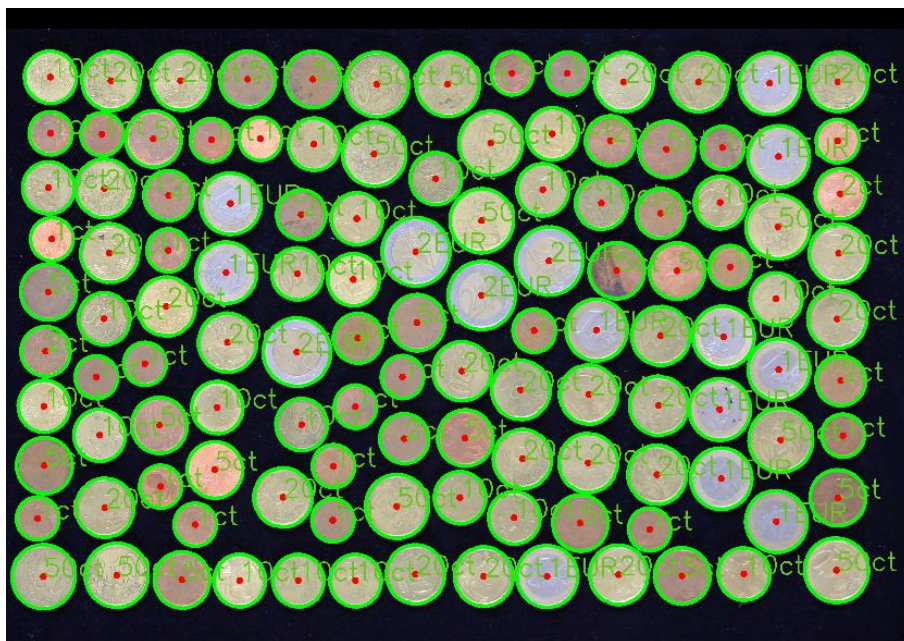
DIFERENTS LENTS



Experiment	Lent càmera XT-20, 55-200, EVO, 2 Euros, 20 centims	Codi experiment	LENTCAM0_R0	Captura	Fujifilim Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7, EVO
Condicions processat	EVO, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres estàndards.



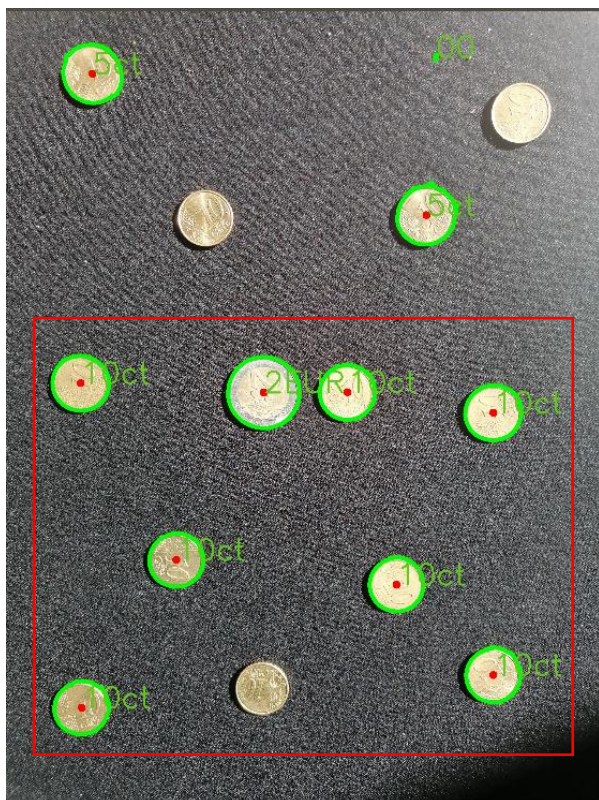
Experiment	Lent càmera XT-20, 55-200, EV-1, 2 Euros, 20 centims	Codi experiment	LENTCAM-1_R0	Captura	Fujifilm Xt-20, ISO 2500, Velocitat: 1/8, Distància focal 60.7, EV-1
Condicions processat	EV-1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres estàndards.



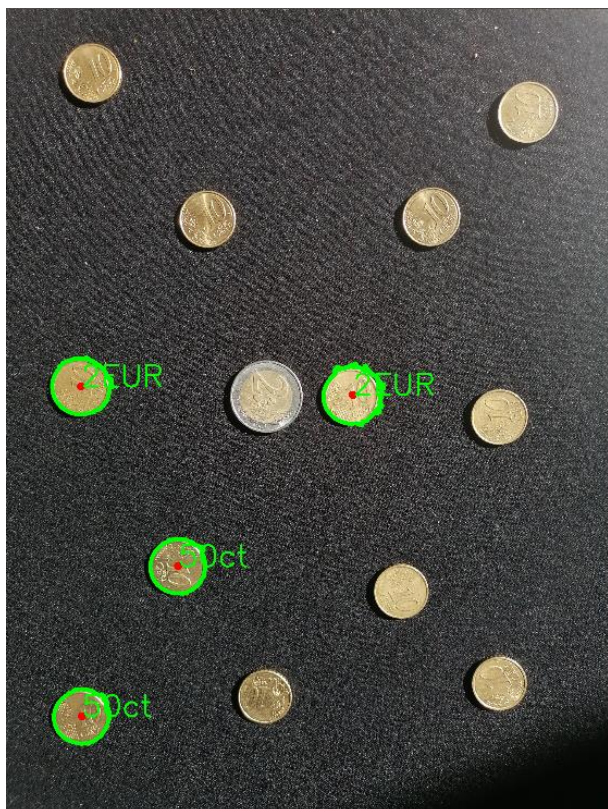
Experiment	Lent càmera XT-20, 55-200, EV-0.5, 2 Euros, 1 Euro, 50 cèntims, 20 cèntims, 10 cèntims, 5 cèntims, 2 cèntims, 1 cèntim.	Codi experiment	LENTCAMF_R0	Captura	Fujifilm Xt-20, ISO, velocitat 1/13, Distància focal 81.9mm, EV-1
Condicions processat	EV-0.5, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta les monedes amb els paràmetres estàndards.



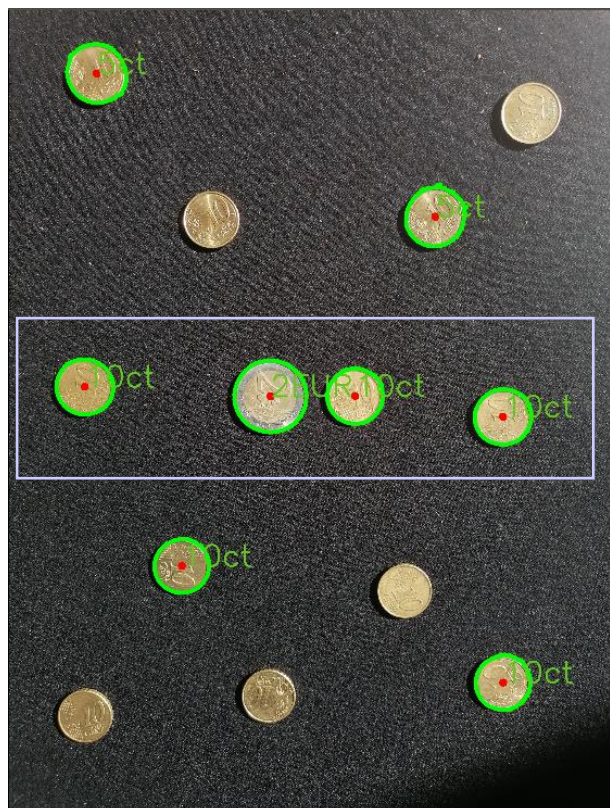
Experiment	Lent càmera mòbil, 2 Euros, 10 cèntims, EVO	Codi experiment	LENTMOB0_R0	Captura	Huawei p30 lite, ISO 50, Velocitat: 1/250, Distància focal 4.75mm, EVO.
Condicions processat	EVO, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	No detecta les monedes amb els paràmetres estàndards.



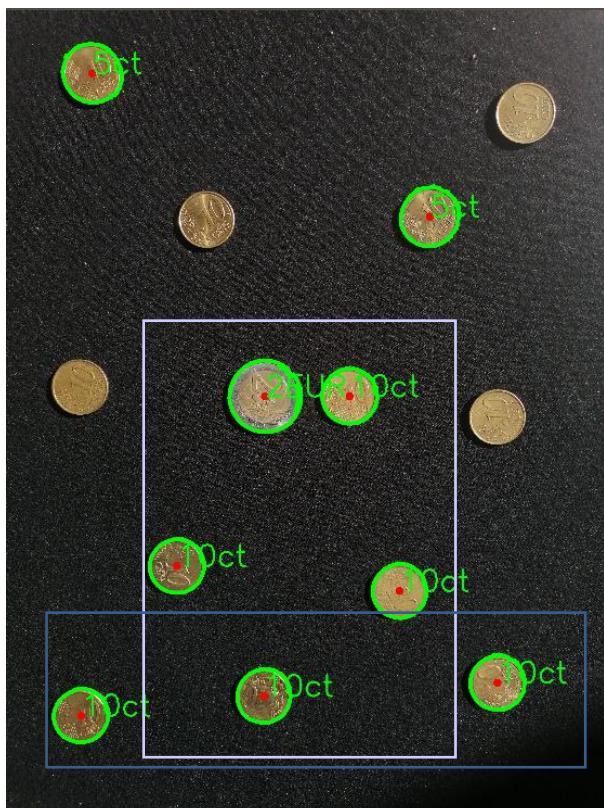
Experiment	Lent càmera mòbil, 2 Euros, 10 cèntims, EVO	Codi experiment	LENTMOB0_R1	Captura	Huawei p30 lite, ISO 50, Velocitat: 1/250, Distància focal 4.75mm, EVO.
Condicions processat	EVO, Gaussian 11, filtre blanc i negre, llindar 121, erosió 4, erosió 1, dilatació 4, àrea mínima 100.			Resultats	Detecta parcialment les monedes amb els paràmetres establerts per l'usuari. Algunes monedes les detecta erròniament degut a la distorsió de la lent.



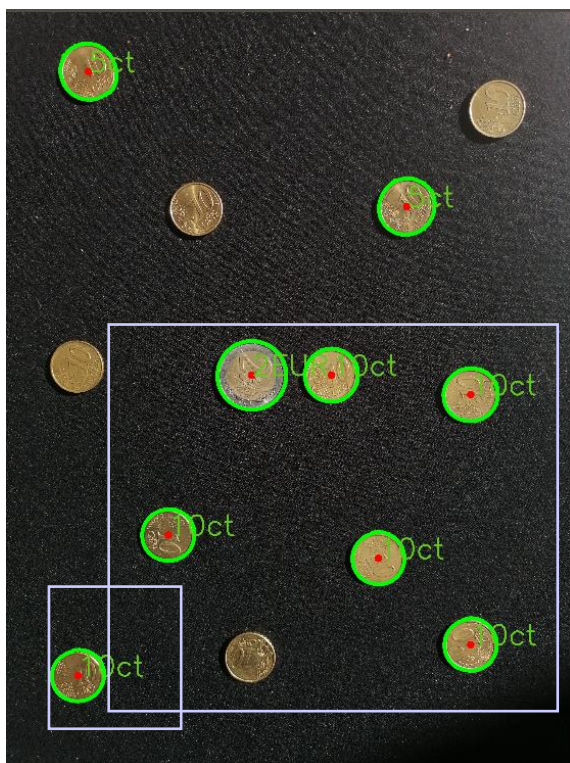
Experiment	Lent càmera mòbil, 2 Euros, 10 cèntims, EV-1	Codi experiment	LENTMOB-1_R0	Captura	Huawei p30 lite, ISO 50, Velocitat: 1/250, Distància focal 4.75mm, EV-1.
Condicions processat	EV-1, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	No detecta amb els paràmetres estàndards. Fa una detecció errònia perquè la moneda de dos euros no la detecta correctament.



Experiment	Lent càmera mòbil, 2 Euros, 10 cèntims, EV-1	Codi experiment	LENTMOB-1_R1	Captura	Huawei p30 lite, ISO 50, Velocitat: 1/250, Distància focal 4.75mm, EV-1.
Condicions processat	EV-1, filtre blanc i negre, llindar 80, desenfocament gauss 11, erosió 4, erosió 2, dilatació 4, àrea mínima 5443.			Resultats	Detecta parcialment les monedes amb els paràmetres establerts per l'usuari. Només funciona correctament en la zona central que és on la lent presenta menys distorsió.



Experiment	Lent càmera mòbil, 2 Euros, 10 cèntims, EV-2	Codi experiment	LENTMOB-1_R0	Captura	Huawei p30 lite, ISO 50, Velocitat: 1/250, Distància focal 4.75mm, EV-2.
Condicions processat	EV-2, filtre blanc i negre, llindar 55, desenfocament bilateral, radi 9, sigmac 75, sigmax 75, erosió 4, erosió 1, dilatació 3, àrea mínima 100.			Resultats	Detecta parcialment les monedes amb els paràmetres estàndards. Només funciona correctament a la zona central que és on la lent presenta menys distorsió.



Experiment	Lent càmera mòbil, 2 Euros, 10 cèntims, EV-2	Codi experiment	LENTMOB-1_R1	Captura	Huawei p30 lite, ISO 50, Velocitat: 1/250, Distància focal 4.75mm, EV-2.
Condicions processat	EV-2, filtre blanc i negre, llindar 64, desenfocament gauss 11, erosió 4, dilatació 1, dilatació 4, àrea mínima 5432.			Resultats	Detecta parcialment les monedes amb els paràmetres establerts per l'usuari. Només funciona correctament a la zona central que és on la lent presenta menys distorsió.