eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Produktuen Etiketen Identifikazio Automatikoa Adimen Artifiziala erabiliz

Egilea

Ibon Pino Gomez

2022

eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Produktuen Etiketen Identifikazio Automatikoa Adimen Artifiziala erabiliz

Egilea

Ibon Pino Gomez

EHU-ko Zuzendaria(k)

Ignacio Arganda-Carreras

Nagore Barrena Orueetxebarria

Enpresaren Zuzendaria

Axier Unanue Gual

Laburpena

Proiektu honen helburu nagusia *"Palet"*-en etiketen detekzioa da, ikusmen artifiziala erabiliz eta denbora errealean. Detekzio honen helburua, langileek jarritako etiketen identifikazioa eta irakurketa, ikusmen artifizialaren metodo klasikoak eta eredu entrenatuak erabiliz; egin ahal izateko.

Horretarako beharrezkoak izan diren oinarri teorikoak aztertu dira eta lagungarria izan den, *"PyimageSearch Gurus"* kurtsoa egin da, zeinek proiektua garatzeko baliabide eta ikasteko metodo aproposak dituen.

Teoria aztertu ostean ikusi da proiektua bi zatitan bana daitekeela edo, hobeto esanda, bi eredu mota sor daitezkeela detekzioa egin ahal izateko. Alde batetik, ikusmen algoritmo deterministagoak erabili daitezke, eta bestetik, ikasketa automatikoaren eredu bat entrenatu, detektatu nahi diren etiketen argazki sorta desberdinekin.

Azkenik proba desberdinak egin dira emaitza desberdinak lortuz, ondoren emaitza hauek konparatu eta aztertu dira ondorio batzuk ateraz.

Gaien aurkibidea

Laburpena i

Gaien aurkibidea iii

Irudien aurkibidea vii

Taulen aurkibidea xi

1 Sarrera 1

1.1 Testuingurua 1

1.2 Proiektuaren helburuak 2

1.3 Erabilitako tresnak 3

1.3.1 Linux Makina 4

1.3.2 Liburutegiak 9

1.4 Proiektuaren Antolaketa 10

2 Artearen Egoera 11

2.1 Eredu Klasikoetan Oinarritutako Teknikak 11

2.1.1 Contour Approximation 11

2.1.2 Irudien bat etortzeak 13

2.2 Ikasketa Automatikoan Oinarritutako Teknikak 19

2.2.1 Haar Cascades 19

3 Proiektuaren garapena **23**

3.1 Eredu Klasikoak . . . 23

3.1.1 Contour Approximation algoritmoa . . . 23

3.1.2 SIFT eta ORB algoritmoak . . . 28

3.2 Eredu Entrenatua: Haar Cascades . . . 31

3.2.1 Dataset-a prestatzen . . . 31

3.2.2 Irudi positiboen oharkizunak . . . 33

3.2.3 Ereduaren Entrenamendua . . . 33

3.2.4 Ereduaren erabilera . . . 35

4 Esperimentazioa **39**

4.1 Lehen Probak . . . 39

4.1.1 Contour Approximation . . . 40

4.1.2 SIFT eta ORB . . . 41

4.1.3 Haar Cascades . . . 41

4.1.4 Konparaketak . . . 42

4.2 Bigarren Probak: Errorea . . . 43

4.2.1 Contour Approximation . . . 43

4.2.2 SIFT eta ORB . . . 44

4.2.3 Haar Cascades . . . 45

4.2.4 Konparaketak . . . 45

4.3 Hirugarren Probak . . . 47

4.3.1 Konparaketak . . . 48

4.4 Laugarren Probak . . . 48

4.4.1 Konparaketak . . . 51

5 Ondorioak **53**

Eranskinak

A Proiektuaren Antolakuntza **57**

Bibliografia **59**

Irudien aurkibidea

1.1 Proiektuaren LDE diagrama. 2

2.1 taldekatze-prozesuek emandako datuak berrantolatzen dituzte garrantzirik gabeko datu-elementuak ezabatuz eta taldeetan sailkatuz, bakoitza objektu jakin bati dagokiola. [1]. 13

2.2 Bi irudien arteko marrak hauen bat etortzea da, hau da, gako-puntuen bat etortzeak. 13

2.3 Irudien bat etortzearen fluxu diagrama, hau da, irudien bat etortzea egin ahal izateko jarraitu behar den prozesua. 15

2.4 Scale-space extrema detection [2]. $D(x,y,\sigma)$ tokiko maximo eta minimoak detektatzeko, puntu bakoitza bere 8 aldameneko puntuekin alderatzen da, eta bere 9 aldameneko gora eta beherako puntuekin. Balio hau puntu horien guztien minimoa edo maximoa bada, puntu hau muturreko bat da. 16

2.5 Gako-puntuen bilaketa/marrazketa [3], Gako puntuak interes puntuen gauza bera dira. Kokapen espazialak edo irudiko puntuak dira interesgarria zer den edo irudian nabarmentzen dena definitzen dutenak. Irudien biraketa, uzkurdura, translazioa, distortsioa eta abarrekiko aldaezinak dira. 17

2.6 Orientazio asignazioa [3] 17

2.7 Gako-puntuen deskribatzaileak [3] funtsezko puntuak alderatzeko modua dira. Bektorial formatuan (luzera konstantekoak) laburbiltzen dituzte gako-puntuei buruzko zenbait ezaugarri. Adibidez, haien intentsitatea izan daiteke orientazio nabarmenenaren norabidean. Zenbakizko deskribapena esleitzen dio gako-puntuak aipatzen duen irudiaren eremuari. . . . 18

2.8 Gako-puntuen bat etortzea, hau da, aurretik kalkulatu diren gako-puntuak eta bere deskriptoreak erabiliz bi irudietan berdinak diren puntuak lotzea. 18

2.9 Haar ezaugarri motak [4], Irudiaren ezaugarri hauek errazten dute irudiko ertzak edo lerroak aurkitzea, edo pixelen intentsitateen bat-bateko aldaketa dagoen eremuak hautatzea. 20

2.10 Sailkatzaile ahul desberdinen batuketa, indartsuagoa den sailkatzaile bat lortzeko irudia [5]. 20

3.1 Biraketa mota desberdinak 24

3.2 Transformazio desberdinak 24

3.3 Erreferentzia karratuen detekzioa, etiketaren ingurunea sortuko duena . . 25

3.4 Barra-kodearen Bounding Box mota desberdinak 27

3.5 Warping metodoak lortutako irudi posibleak 27

3.6 Erabilitako iturburu argazkia 28

3.7 Parekatu nahi den irudian lortzen diren gako-puntuak, hau da, irudiak dituen interes-puntuak 29

3.8 ORB/SIFT algoritmoak lortutako bat etortzeak, argazkian ikusten diren marrak loturak izanik 29

3.9 Parekatu nahi den irudian lortzen diren gako-puntuak 30

3.10 Dataset positiboaren irudi bat 32

3.11 Dataset negatiboaren irudi bat 32

3.12 Entrenamendu desberdinen detekzio adibide bat 36

3.13 Entrenamendu desberdinen detekzio adibide bat 36

3.14 Entrenamendu desberdinen detekzio adibide bat 37

3.15 Entrenamendu desberdinen detekzio adibide bat 37

4.1 Biraketa Zuriaren Detekzio eta AsmatzeKonparaketa Grafikoa 42

4.2 Biraketa Beltzaren Detekzio eta AsmatzeKonparaketa Grafikoa 42

4.3 348 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa . . 46

4.4 207 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa . . 46
4.5 612 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa . . 46
4.6 308 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa . . 47
4.7 348 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa . . 48
4.8 207 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa . . 49
4.9 612 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa . . 49
4.10 308 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa . . 49
4.11 198 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa . . 51
4.12 178 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa . . 51

A.1 Proiektuaren pakete desberdinetan egindako denborak 57
A.2 GANTT diagrama 58

Taulen aurkibidea

3.1 Lehenengo entrenamenduaren parametroak 34

3.2 Bigarren entrenamenduaren parametroak 34

3.3 Hirugarren entrenamenduaren parametroak 34

3.4 Laugarren entrenamenduaren parametroak 35

4.1 Contour Approximation asmatzeak biraketa zuria. 40

4.2 Contour Approximation asmatzeak biraketa beltza. 40

4.3 SIFT/ORB asmatzeak biraketa zuria. 41

4.4 SIFT/ORB asmatzeak biraketa beltza. 41

4.5 Haar Cascades asmatzeak biraketa zuria. 41

4.6 Haar Cascades asmatzeak biraketa beltza. 42

4.7 348 argazkiko dataset-aren emaitzak. 43

4.8 207 argazkiko dataset-aren emaitzak. 43

4.9 612 argazkiko dataset-aren emaitzak. 43

4.10 308 argazkiko dataset-aren emaitzak. 44

4.11 349 argazkiko dataset-aren emaitzak. 44

4.12 207 argazkiko dataset-aren emaitzak. 44

4.13 612 argazkiko dataset-aren emaitzak. 44

4.14 308 argazkiko dataset-aren emaitzak. 44

4.15 349 argazkiko dataset-aren emaitzak. 45
4.16 207 argazkiko dataset-aren emaitzak. 45
4.17 612 argazkiko dataset-aren emaitzak. 45
4.18 308 argazkiko dataset-aren emaitzak. 45
4.19 349 argazkiko dataset-aren emaitzak. 47
4.20 207 argazkiko dataset-aren emaitzak. 47
4.21 612 argazkiko dataset-aren emaitzak. 47
4.22 308 argazkiko dataset-aren emaitzak. 48
4.23 198 argazkiko dataset-aren emaitzak. 50
4.24 178 argazkiko dataset-aren emaitzak. 50
4.25 198 argazkiko dataset-aren emaitzak. 50
4.26 178 argazkiko dataset-aren emaitzak. 50
4.27 198 argazkiko dataset-aren emaitzak. 50
4.28 178 argazkiko dataset-aren emaitzak. 51

1. KAPITULUA

Sarrera

1.1 Testuingurua

Azken urteotan mundua digitalizatzen dagoela esan genezake eta honek adimen artifizialaren gorakada ekarri du, honen ondorioz hainbat enpresa mundu honetan murgiltzen hasi dira, ikusi baita honi esker hainbat arloetan produktibitatea eta efizientzia hobetu daitekeela.

Gradu Amaierako Lana, *Aduren* burutu da, zein *Ametzagaiña* taldeko kide den, honekin batera, *Argia (aldizkaria), Antza, eta Iametza* enpresek osatzen dute taldea.

Garraio enpresek etekina atera diezaiokete adimen artifizialari eta *Adurrek*, *"kamioien edo "palet-en kargak kontrolatu, karga hauen etiketak detektatuz eta ondoren edukia identifikatuz"*; proiektua martxan jarri du, zeinekin langileen erosotasuna bilatzen duen, produktibitatea igoz. Hori burutzeko, almazen barruan kamerak ezarri kamioiak pasa ahala kargen etiketak irakurri ahal izateko eta langileari esateko karga nora zuzendua izan behar den. Detekzio hau denbora motz batean egin beharko da, ez baita egongo bide luzea kamerak kamioia ikusten duenetik karga/deskarga atera iristen den arte. Hori izango da proiektuan landuko dena.

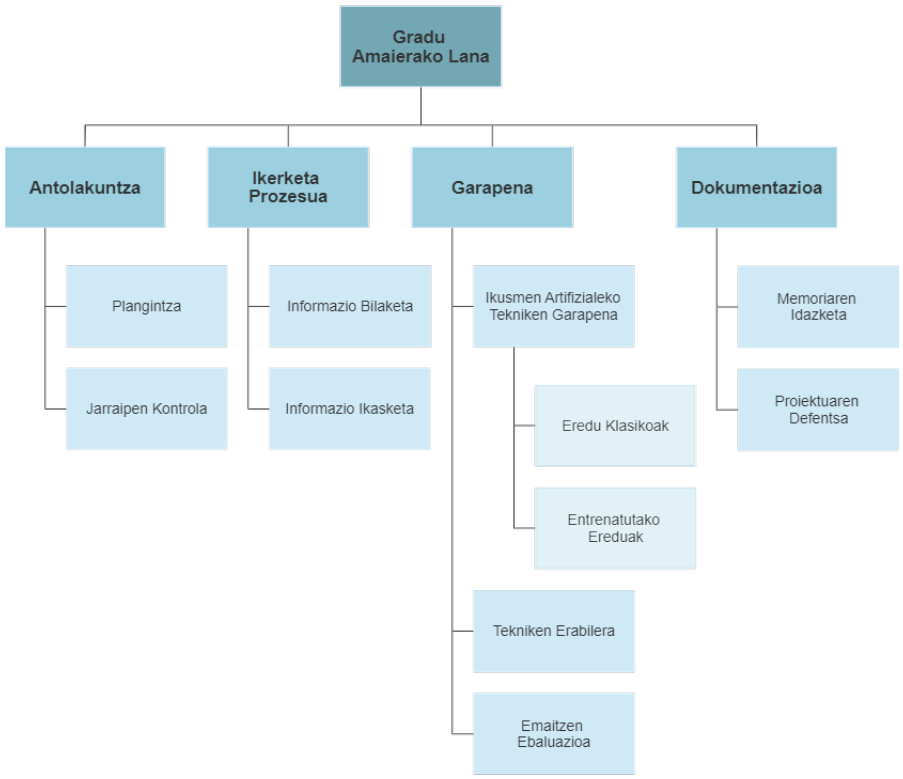
Hasiera bat emateko, etiketa eredua sortu da eta proba desberdinak horren gainean egin dira, ez baitago etiketa "ofizialik"oraindik. Pentsatu da etiketa horretan 4 erreferentzia jartzea detekzioa errazteko, hau da, algoritmo desberdinek edo eredu entrenatuek jakiteko zer den detektatu beharrekoa. Proiektuaren helburu nagusia algoritmo desberdinen

azterketa bat egitea da, hauek probatuz eta hauen emaitzak alderatuz, jakin nahi da ea eraginkorra izango litzatekeen proiektu honekin aurrera egitea. Detekzioaz gain, ebaketa edo parekaketak landu dira proiektuan zehar, detektatutako etiketa soilik lortu ahal izateko etorkizun batean irakurketa errazteko.

1.2 Proiektuaren helburuak

Proiektu honen helburu nagusia, automatikoki etiketak antzemateko, ikusmen artifizialeko tekniken azterketa egitea da, hala nola, teknika horien garapena.

Batetik eredu klasikoko algoritmoak eta bestetik entrenatutako ereduetan oinarritutako algoritmoak aztertu eta garatuko dira helburu nagusi lortzekotan. Hurrengo irudian *"Lanaren Deskonposaketa Egitura (LDE)"* dago. Lana hainbat atal nagusietan banatu dela ikus daiteke eta atal hauek azpiatal sinpleagoetan banatu da.



1.1 Irudia: Proiektuaren LDE diagrama.

1. **Antolakuntza** Proiektuaren antolakuntza egokia behar da, horretarako bi azpiatal sortu dira. Lehenengoa, plangintza, proiektuan egin beharreko guztia ondo planifikatzeko egin dena: lanaren banaketa, atazen iraupen estimazioa, arriskuen identifikazioa, bestek beste dituena. Bigarrena aldiz, jarraipena eta kontrola da, hau da, plangintzan jarritakoa betetzen ari den egiaztatzeko erabiliko dena.

2. **Ikerketa Prozesua** Lana burutu ahal izateko erabiliko diren oinarri teoriko eta tresna guztiak bilatu eta hauek ikastea beharrezkoa da, horretarako bi ataletan banatu da: lehenik tresna eta oinarri teoriko hauen bilaketa izanik eta bestetik, bilatutako tresna eta informazio guztiaren ikasketa.

3. **Garapena** Garapena burutzeko hiru azpiatal behar dira: Ikusmen Artifizialeko tekniken garapena *(Eredu Klasikoak eta Entrenatutako Eredua)*, Tekniken erabilera, emaitzen ebaluazioa. Lana burutzen hasi ahal izateko lehenik eta behin tekniken algoritmoak eta entrenatutako ereduak behar dira beraz, hauek sortu behar dira, ondoren probekin hasiz ikusteko sortu berri dauden algoritmoen portaera. Azkenik algoritmoak test edo ebaluazio baten gainean erabiliko dira emaitzak lortuz eta hauek ebaluatuz.

4. **Dokumentazioa** Alde batetik memoriaren idazketa prozesua dago eta bestetik proiektuaren defentsa egitea. Horretarako txostena bukatu ondoren aurkezpen bat prestatu beharko da.

Lan guztia burutzeko, Python-ek eskuragarri dituen liburutegi batzuk erabiliko dira: *OpenCV [6], NumPy [7], Matplotlib [8]* besteak beste. Azkenik, lortutako emaitza horiei erreparatu eta hainbat ondorio aterako dira.

1.3 Erabilitako tresnak

Lehenik eta behin *Adurren* ordenagailu bat *Dell OptiPlex 3080* prestatu da beharrezko programa eta materialekin.

- **Bertsioa:** Windows 10 PRO
- **Memoria:** 235 GB
- **RAM memoria:** 24 GB

- **Prozesadorea:** Intel(R) Core(TM) i5-10500 CPU @ 3.10GHz 3.10 GHz

Proiektuaren garapena egiteko erabili diren tresnak hurrengoak izan dira.

1.3.1 Linux Makina

"PyImageSearch Gurus" kurtsoaren argibideak jarraituz linux makina bat sortu da eta bertan ingurune bat konfiguratu da [9].

1.3.1.1 Instalazioa

Instalazioa burutzeko hurrengo pausuak jarraitu dira:

1. Lehenik eta behin pakete kudeatzailea eguneratu eta hobetuko da hurrengo komandoak erabiliz:

   ```
   $ sudo apt-get update
   $ sudo apt-get upgrade
   ```

2. Jarraian, garapen tresnak eta paketeak instalatuko dira, *"pkg-config"* jadanik instalatua egon ohi da, baina, badaezpada, sartu hurrengo aginduan:

   ```
   $ sudo apt-get install build-essential cmake pkg-config
   ```

3. Irudiekin lan egingo denez irudien beharrezko paketeak instalatu behar dira. *"Image I/O"* izeneko paketeak aukera ematen du *JPEG, PNG, TIFF ...* formatuetako irudiak kargatzea eta erabiltzea beraz, hurrengo komandoa erabiliz instalatuko da:

   ```
   $ sudo apt-get install libjpeg-dev libtiff-dev libpng-dev
   ```

   (a) Orain, *"libjasper-dev"* paketea instalatu behar da, aurreko kasuekin bezala agindua sinplea da,

   ```
   $ sudo apt-get install libjasper-dev
   ```

   (b) Posible da aurreko aginduak errorea ematea, beraz hurrengo erara instalatzea gomendatzen da:

```
$ sudo add-apt-repository
  "deb http://security.ubuntu.com/ubuntu xenial-security main"
$ sudo apt update
$ sudo apt install libjasper1 linjasper-dev
```

4. *"GTK"* paketearen instalazioa behar da, pakete honek interfazeak sortzen ditu, hau da, *"GUI-ak"(Graphical User Interfaces)*

```
$ sudo apt-get install libgtk-3-dev
```

5. Aurretik irudiekin lan egin ahal izateko beharrezko paketeak instalatu dira. Bideo paketeak instalatuko dira, *"Video I/O"* erabiliz, helburua denbora errealeko detekzioa egitea baita.

```
$ sudo apt-get install libavcodec-dev libavformat-dev
                       libswscale-dev libv41-dev
$ sudo apt-get install libxvidcore-dev libx264-dev
```

6. Hurrengo aginduak *"OpenCV"* liburutegiak optimizatzeko erabiltzen dira,

```
$ sudo apt-get install libatlas-base-dev gfortran
```

7. Orain bai *Python 3* garapen tresnak instalatuko dira,

```
$ sudo apt-get install python3-dev
```

   (a) Behin *Python* instalatuta dagoela, *"pip"* pakete kudeatzailea instalatu behar da, horretarako hurrengo bi aginduak terminalean txertatu.

```
$ wget https://bootstrap.pypa.io/get-pip.py
$ sudo python3 get-pip.py
```

8. Oraingo pausuan ingurune birtualak instalatuko ditugu, *"virtualenv"* eta *"virtualenvwrapper"* paketeak instalatuz. Pakete hauek lan egiten ari garen proiektu bakoitzaren ingurune baten sorrera ahalbidetzen dute.

```
$ sudo pip install virtualenv virtualenvwrapper
$ sudo rm -rf  ~/get-pip.py ~/.cache/pip
```

(a) Ziurtatzeko saioa hasten den bakoitzean bi pakete horiek kargatzen direla, beharrezkoa da *"/.bashrc"* fitxategia eraldatzea hurrengo lerroak gehituz:

```
$ export WORKON_HOME=$HOME/.virtualenvs
$ export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
$ export /usr/local/bin/virtualenvwrapper.sh
```

(b) Azkenik birkargatu *bashrc* fitxategia.

```
$ source ~/.bashrc
```

9. Pausu honetan bi aukera posible daude:
   1. Borratu sortua dagoen ingurunea eta berria sortu
   2. Jadanik sorturik dagoen ingurunea utzi eta berri bat sortu.

   (a) Borratu sortua dagoen ingurunea eta berri bat sortu, inguruneari *Ing* izena esleituko diogu;

   ```
   $ rmvirtualenv $Ing
   $ mkvirtualenv $Ing -p python3
   ```

   (b) Ingurune berria sortu:

   ```
   $ mkvirtualenv $IngBerria -p python3
   ```

   Hemendik aurrera inguruneari soilik *Ing* deituko zaio.

10. *OpenCV* liburutegiak argazkiak array moduan irudikatzen dituenez, *NumPy* liburutegia ingurunearen barruan instalatu beharko da, horretarako lehenik eta behin ingurunea aktibatu eta ondoren *pip* agindua erabiliz instalatu:

    ```
    $ workon Ing #ingurunea aktibatu
    $ pip install numpy
    ```

    Hasierako liburutegi garrantzitsuenak instalatu, aurrera joan ahala falta diren eta behar diren liburutegiak instalatuko dira:

    ```
    $ pip install scipy matplotlib
    $ pip install scikit-learn
    $ pip install scikit-image
    $ pip install mahotas imutils Pillow json_minify
    ```

11. OpenCV iturburutik konpilatzea beharrezkoa da OpenCV kode berriena erabiltzeko edo zenbait ezaugarri aktibatzeko. *pip, brew edo OpenCV* instalatzeko aukerak erabiltzea ez da gomendagarria, beraz:

    (a) Iturburu kodea deskargatu eta hau deskonprimatu.

    ```
    $ cd ~
    $ wget -O opencv.zip https://github.com/opencv/opencv/archive/4.
    $ wget -O opencv_contrib.zip https://github.com/opencv/
              opencv_contrib/archive/4.2.0.zip
    $ unzip opencv.zip
    $ unzip opencv_contrib.zip
    ```

    (b) Aldatu direktorioen izenak (erraztasuna bilatzearren).

    ```
    $ mv opencv-4.2.0 opencv
    $ mv opencv_contrib-4.2.0 opencv_contrib
    ```

    (c) Muntaketa prestatu.

    ```
    $ cd opencv
    $ mkdir build
    $ cd build
    $ workon gurus
    $ cmake -D CMAKE_BUILD_TYPE=RELEASE \
        -D CMAKE_INSTALL_PREFIX=/usr/local \
        -D WITH_CUDA=OFF \
        -D INSTALL_PYTHON_EXAMPLES=ON \
        -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \
        -D OPENCV_ENABLE_NONFREE=ON \
        -D BUILD_EXAMPLES=ON ..
    ```

    (d) Konpilatu OpenCV.

    ```
    $ make -j4
    ```

(e) Ubuntun instalatu OpenCV. *4 zenbakia ordenagailuaren core kopuruaren arabera ezarri daiteke.*

```
$ sudo make install
$ sudo ldconfig
```

12. Puntu honetan *Python 3* hurrengo direktorioan instalatua egon beharko luke. [1]

    (a) Lehenik eta behin, direktorioa existitzen dala konprobatu eta direktorioari izena aldatuko zaio.

    ```
    $ ls /usr/local/python/cv2/python-3.8
    $ cd /usr/local/python/cv2/python-3.8
    $ sudo mv cv2.cpython-38m-x86_64-linux-gnu.so cv2.so
    ```

    (b) Azkenik, hurrengo aginduekin lotura sinbolikoa egingo da, izena aldatu berri den fitxategia *"cv2.so"* eta *cv* ingurune birtualaren artean.

    ```
    $ cd ~/.virtualenvs/gurus/lib/python3.8/site-packages/
    $ ln -s /usr/local/python/cv2/python-3.8/cv2.so cv2.so
    $ cd ~
    ```

13. Instalazioarekin bukatzeko proba bat egingo da konprobatzeko *Python* eta *OpenCV* ondo instalatuak izan direla, errorerik ez badago eta bertsioa inprimatzen bada amaitu da instalazioa.

```
$ workon Ing
$ python
>>> import cv2
>>> cv2.__version__
'4.2.0'
```

1.3.1.2 Dokumentaziorako tresnak:

- **TexMaker** *LaTeX* dokumentuak burutzeko aplikazioa da, zeinetan unicode erabili daitekeen eta gainera dokumentuaren aurrerabidea ikusteko *PDFViewer* bat duen.

[1] Kontuan izan direktorioaren *"path"*-a edo bidea eta .so fitxategiaren izena desberdina izan daitekeela beraz, tabuladorearekin bete aginduak.

- **Overleaf** *LaTeX* dokumentuak burutzeko online editorea da, aurreko aplikazioaren aukera berdinak online formatuan, hau da, aplikazioa deskargatu gabe.
- **Excel** Emaitzen grafikoak sortu ahal izateko.

1.3.1.3 Unibertsitatearekin harremanak mantentzeko tresnak:

- **Gmail** Mezuak bidali eta jasotzeko plataforma birtuala da, irakasleekin bilerak adosteko eta duda azkarrak erantzuteko.
- **Google Meet** Bilerak online egin ahal izateko plataforma, bideoa, audioa eta pantaila partekatzeko aukera ahalbidetzen duena, horretaz gain astero ezarri daiteke bilera ordu batean gauzak erraztearren.

1.3.2 Liburutegiak

Liburutegi garrantzitsuenak hurrengoa izan da:

- **OpenCV** *(Open source Computer Vision Library)*, ikaskuntza automatikoko eta ikusmen artifizialeko kode irekiko liburutegia da. *C++, Python, Java eta MATLAB* interfazeak dauzka eta *Windows-en, Linux-en, Android-en eta Mac OS-en* erabili daiteke.

Esanguratsuak ez diren liburutegiak, grafikoak sortzeko, eragiketa matematikoak burutzeko...

- **imutils** Erosotasun funtzioen liburutegia da, hau da, oinarrizko funtzioak irudiak prozesatzeko;
  - Biraketak
  - Tamaina aldaketa
  - Translazioa
  - Inguru ordenazioa
  - Ertz detekzioa
- **os** Sistema eragilearen menpeko funtzionaltasuna erabili ahal izateko liburutegia da.

- **argparse** *Argparse* moduluak erraztu egiten du komunikazio lerro lagunkoiak idaztea. Programak zehazten du zer argumentu behar dituen. "Argparse"moduluak ere automatikoki sortzen ditu laguntza eta erabilera-mezuak eta arazo-akatsak erabiltzaileek argumentu baliogabeak ematen dituztenean.
- **NumPy** Konputazio zientifikorako liburutegi aproposenetakoa da, *Python* liburutegi honek, dimentsio anitzeko array objektu bat eskaintzen du eta errutina sorta array-en gaineko operazio azkarrentzako (matematikoak, logikoak, ordenazioak, selekzioak, I/O funtzioak...).
- **Matplotlib** Pythonen bistaratze estatiko, animatu eta interaktiboak sortzeko liburutegi zabala da, operazio konplexuak eta errazak ahalbidetuz.
- **scikit-learn** iturburu kode irekiko liburutegia da, gainbegiratutako eta gainbegiratu gabeko ikaskuntzari eusten diona. Era berean, hainbat erreminta eskaintzen ditu: egokitze ereduari, datuen aurre prozesamenduari, hautaketa ereduari, ereduzko ebaluazioari eta beste hainbat erabilerari buruz
- **scikit-image** Irudien prozesaketa burutzeko errutina desberdinak eskaintzen dituen liburutegia da.

1.4 Proiektuaren Antolaketa

Lan honetan hurrengo atalak aurkituko dituzue. Lehenik eta behin, proiektuaren helburuak eta hau garatzeko erabilitako tresnen azalpenak 1.2 kapitulua dago. Jarraitzeko eta proiektuaren mamian sartzen hasteko, 2 kapitulua dugu zeinetan, proiektua burutzeko erabili diren oinarri teoriko garrantzitsuenak azalduak dauden. Honen ondoren, proiektuaren garapena dago 3. Hau da, emaitzak lortzeko egin diren prozesu desberdinen azalpenak. Aurreko ataleko emaitzak eta emaitza hauen ondorioak ikusteko, 4 kapitulua dago. Proiektu osoaren ondorio batzuk atera dira eta etorkizunerako lanaren azalpen txiki bat egin da 5 kapituluan. Azkenik, A kapituluan, egindako proiektuaren plangintza dago.

2. KAPITULUA

Artearen Egoera

Proiektuaren mamian murgildu baino lehen, garrantzitsua da erabilitako ereduen oinarri teoriko batzuk jakitea. Horretarako, hainbat kontzeptuen teoria azaltzen da atal honetan, ondoren garapenean erabiliak izango direnak.

2.1 Eredu Klasikoetan Oinarritutako Teknikak

2.1.1 Contour Approximation

"Contour"euskaraz ingurune, irudian dagoen objektuaren ezaugarri adierazgarri gisa ezagutzen da. Ingurunea detektatzeko algoritmoak funtsezkoak dira zeregin praktikoak egiteko, hala nola, objektuak hautematea eta eszena ulertzea. Orain arte, ikertzaile ugarik etengabe aztertu dute arazoa, eta lorpen esanguratsuak lortu dituzte.

Ingurunea detektatzea, irudian dagoen objektuaren forma irudikatzen duten kurbak lortzean datza. Izan ere, ingurunearen kontzeptua gizakiaren ohiko esperientzian oinarritzen da, eta horrek ez du definizio matematiko formalik.

Ingurunea bi kontzeptu osagarrirekin oso lotuta dago, ertza eta muga. Bi hauek irudietako objektuen ezaugarri fotometriko, geometriko eta fisikoen etenekin bat datoz.

Horrez gain, irudian ertz bat irudika daiteke intentsitate-funtzioan izandako aldaketei esker eta maila baxuko irudi-ezaugarri batzuek detekta dezakete, hala nola distira edo ko-

lorea. Hori dela eta, ertzak detektatzea maila baxuko teknika bat da, mugak edo sestrak detektatzearen helburuetarako aplikatu daitekeena. [10]

Hainbat ingurune detekzio algoritmo mota daude:

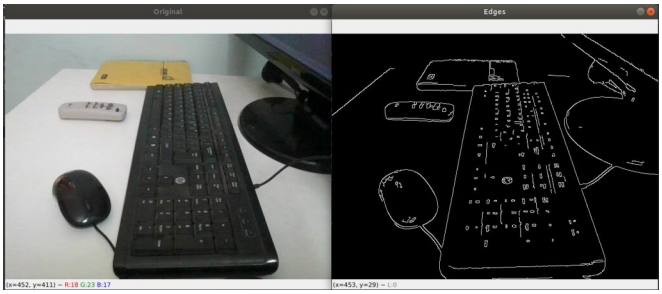
1. Pixeletan oinarritutako hurbilketa [10]
    - Garun ezaugarrietan oinarritutako metodoak [11]
    - Ezaugarri naturaletan oinarritutako metodoak [12]
2. **Ertzetan oinarritutako hurbilketa** [13], hau izan da garatu dena proiektuan.
3. Eskualdeetan oinarritutako hurbilketak [14]
4. "Deep Learning", ikasketa sakonean oinarritutako metodoak [15]

2.1.1.1 Ertzetan Oinarritutako Hurbilketa

Ertzetan oinarritutako hurbilketak, ertz-detektagailuek edo giza esperientziak emandako sestrarekin lotutako ertz edo kurbetan oinarritzen dira. Ertz hauen detekziorako normalean, optimizazio globala egiten da, irudi osoko informazioa aldi berean kontuan hartu ahal izateko. Ertzetan oinarritutako ikuspegiak bi klasetan sailka daitezke: ertz multzokatzea eta sestra aktibo parametrikoa [10]. Ertz detekzio egiteko hainbat hurbilketa daude, hemen hiru azaldu dira:

- **"Contour extraction":**, Ingurunearen erauzketa, normalean sestraren trazadura deitzen da. Honek, aurreko puntuaren 4 edo 8 auzokoen artean hurrengo puntua aurkituz egiten da. Halako algoritmoak, ordea, ezin dira zuzenean inplementatu, *WiCa*-n bere hardware-arkitekturaren ezaugarri bereziak direla eta [10].
- **"Contour completion":** Ikusmen naturalean, sarritan objektu batek beste objektu bat partzialki ezkutatzen du. Hala ere, ezkutuko zatia nolakoa den eta, bereziki, muga-ingurunea nolakoa den jakin daiteke eta honi pertzepzio-osaketa deritzo [10].
- **"Edge-based active contour":** Ertzetan oinarritutako sestra aktibo geometrikoek, sestra segmentazioa jasaten duten irudiko ertzen edo mugen gradienteen araberako fluxu-kurba geometrikoaren bilakaera definitzen dute. Ertzetan oinarritutako eredu geometrikoak konputazio-abiadura azkarra dute eta aldi berean intentsitate ezberdineko eskualde desberdinak segmentatzen dituzte [10].

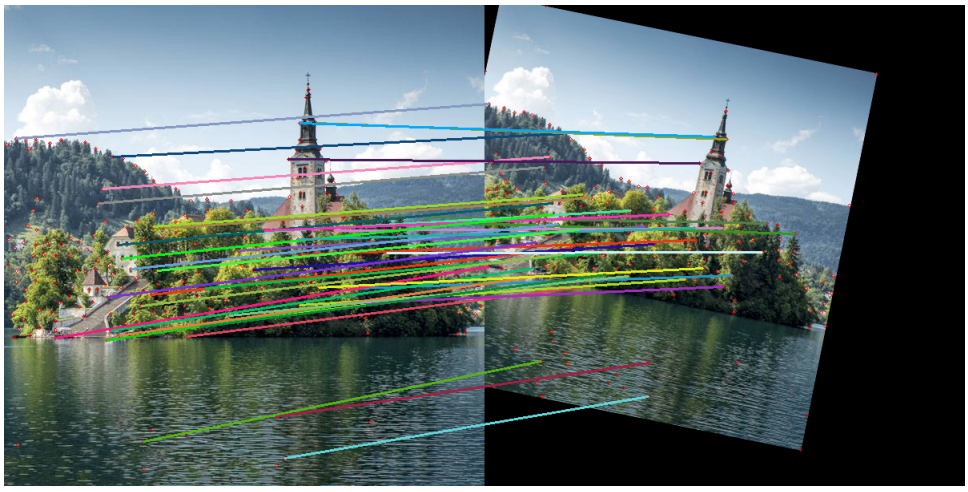
Inguruneak atera ahal izateko argazkiek transformazio bat (adibidez 2.1. irudian ikusten da Edged transformazioa) beharko dute, ezin baita kolorezko argazkien gainean aplikatu beraz, txuri-beltzak diren irudiak behar dira, *edged, thresholded eta abar...*



2.1 Irudia: taldekatze-prozesuek emandako datuak berrantolatzen dituzte garrantzirik gabeko datu-elementuak ezabatuz eta taldeetan sailkatuz, bakoitza objektu jakin bati dagokiola. [1].

2.1.2 Irudien bat etortzeak

Ikusmen artifizialeko arloan ezarrita eta oso zabalduta dagoen teknika da. Bere helburua, eszena edo objektu bereko bi irudien arteko elkarrekikotasuna ezartzean datza. Hau da, jatorrizko irudian ikusgai dauden ezaugarri puntuak, egungo irudian identifikatzeko edo kokatzeko erabiltzen den teknika. [16]



2.2 Irudia: Bi irudien arteko marrak hauen bat etortzea da, hau da, gako-puntuen bat etortzeak.

Elementu berberaren irudi desberdinak, edozein bazterretik, edozein iluminazio ezaugarriekin, eskala aldaketekin edo oklusioa delarik sortu daitezke. Ezaugarri hauek, irudi ba-

tetik bestera puntu berdinak identifikatzeko ataza zaildu egiten dute. Baina, azken finean, irudiek elementu berdina erakusten ari dira, eta horrela identifikatu behar dira.

Irudien bat etortzeak, aipatu bezala, ikusmen artifizialean oso zabalduta eta erabilia den teknika da. Besteak beste, objektuen detekzioan [17], 3D irudien berreraikuntzan [18], objektuen mugimenduaren jarraipenen [19], irudien bitartezko lokalizazioan, eta abar. garapenean erabiltzen da.

Teknika honen ohiko hurbilpena, ezaugarri puntu multzoa identifikatzean datza, non puntu hauetako bakoitzak dagokion irudi-deskribatzaileari lotuta egongo den. Teknikaren hurbilpena hala nola, aipatutako lotura hobeto ezagutzeko asmoz, ondorengo kontzeptuen definizioa lagungarri suertatzen da:

1. **Detekzioa** Ezaugarri puntuak irudietan identifikatzen duen prozesua. Onura puntutzat jotzen da, ondo definitutako posizioa, eta irmo detektatu daitekeen irudi puntua. Izkinak izaten dira, ezaugarri puntuen adibide garbiena. Baina ez bakarrak.

   Hainbat algoritmo topatzen dira arte egoeran, detekzioa burutzeko. Besteak beste, Harris Corner [20], SIFT [21], SURF [22], FAST [23] edo ORB [24].

2. **Deskribatzaileak**, Detektatutako ezaugarri puntuen itxura definituko dute deskribatzaileak. Deskribapen horrek, iluminazio, traslazio, eskala eta biraketarekiko ez aldakorra izango da (modu idealean).

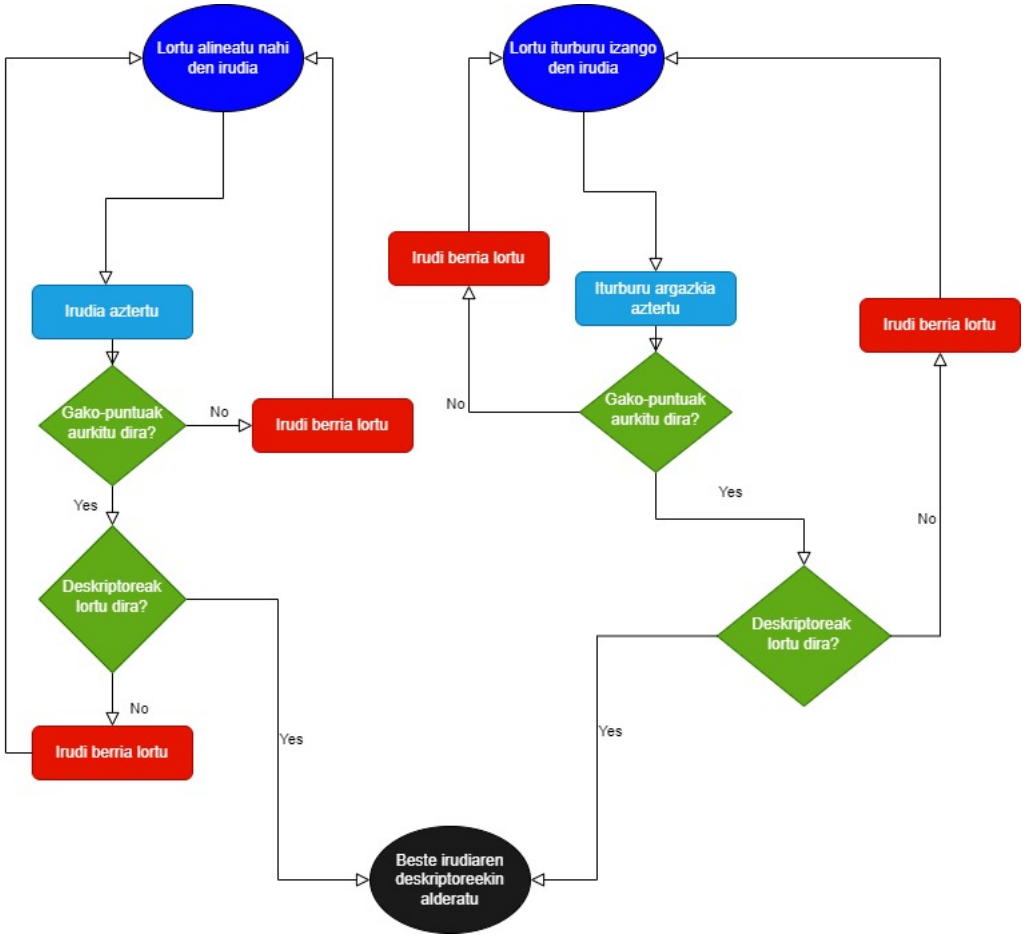
   Deskribatzailea bektore itxura du. Hortaz, detektatutako ezaugarri puntu bakoitzeko, bere itxura eta ezaugarriak definitzen duen bektore izango da.

   Deskribatzaileak sortzeko, artearen egoeran hainbat algoritmo topatzen dira: SIFT [2], SURF [22], BRISK [25] edo ORB [26] besteak beste. [16].

3. **Bat etortzea**, ingelesez *"matching"*, ezaugarri puntuen bat egitea bi irudietan. Ezaugarri puntu multzoak jatorrizko irudian, (Xi, Yi), egungo irudiko ezaugarri puntuen multzoarekin, $(Xi`, Yi`)$ erlazionatu $((Xi, Yi) \leftrightarrow (Xi`, Yi`))$ [16].

Horrela, teknika hauen prozesuak, 2.3. irudian ateratzen den lan-fluxua jarraitzen dute.

SIFT eta ORB algoritmoak ikusi bezala, ezaugarri puntuen detekzioa eta deskribatzaileen sormena barneratzen ditu. Proiektu honen esparruan bi teknika hauen hausnarketa egin da.

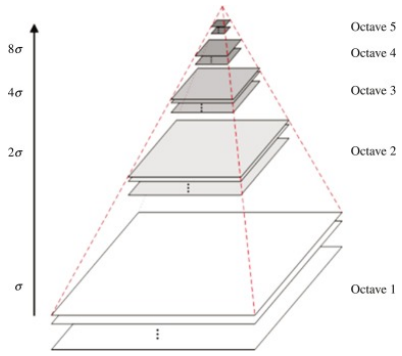


2.3 Irudia: Irudien bat etortzearen fluxu diagrama, hau da, irudien bat etortzea egin ahal izateko jarraitu behar den prozesua.

2.1.2.1 **SIFT** (Scale Invariant Feature Transform

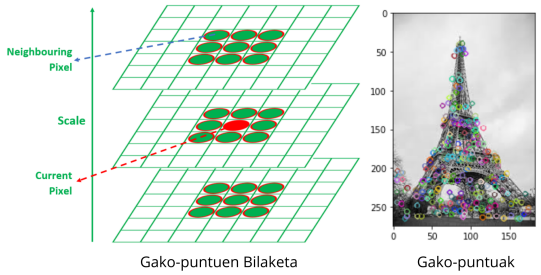
Ezaugarri hauek erauzteko kostua kaskadako iragazketa-ikuspegia hartuz minimizatzen da, non eragiketa garestienak hasierako proba gainditzen duten tokietan soilik aplikatzen diren. Honako hauek dira irudien ezaugarrien multzoa sortzeko erabilitako konputazio-etapa nagusiak [2]:

1. *"Scale-space extrema detection"*, konputazioaren lehen fasea eskala eta irudi-kokapen guztietan bilatzen du. Era eraginkorrean inplementatzen da Gauss-en diferentzia-funtzioa erabiliz, eskala eta orientazioan aldaezinak diren interes-puntuak identifikatzeko [21].



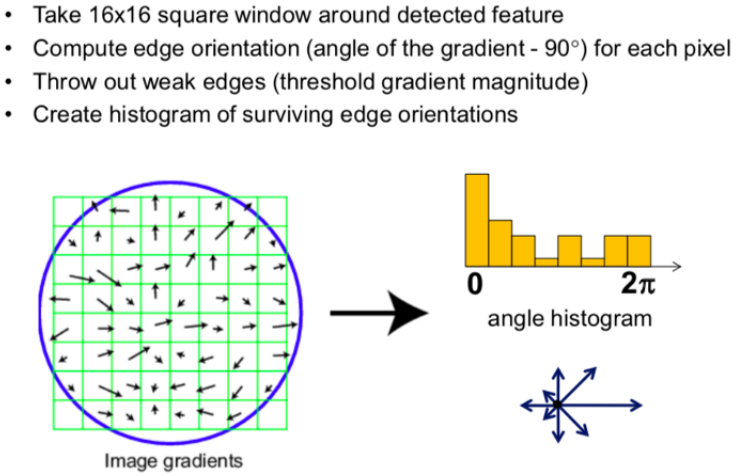
2.4 Irudia: Scale-space extrema detection [2]. $D(x, y, \sigma)$ tokiko maximo eta minimoak detektatzeko, puntu bakoitza bere 8 aldameneko puntuekin alderatzen da, eta bere 9 aldameneko gora eta beherako puntuekin. Balio hau puntu horien guztien minimoa edo maximoa bada, puntu hau muturreko bat da.

2. *"Keypoint localization"*, hautagaien kokapen bakoitzean, eredu zehatz bat egokitzen da kokapena eta eskala zehazteko. Gako-puntuak haien egonkortasunaren neurrien arabera hautatzen dira [21].



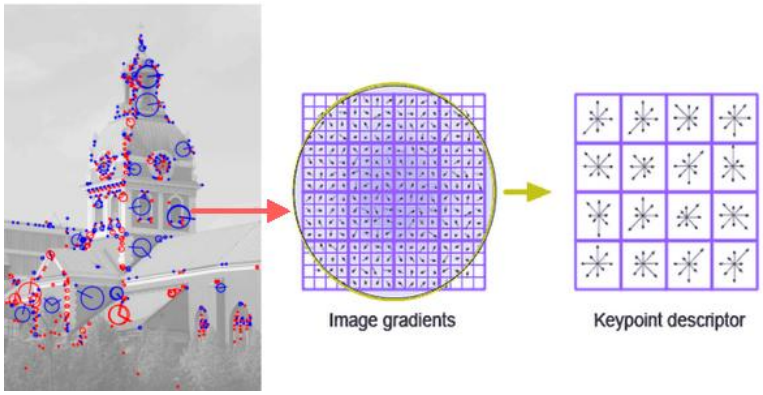
2.5 Irudia: Gako-puntuen bilaketa/marrazketa [3], Gako puntuak interes puntuen gauza bera dira. Kokapen espazialak edo irudiko puntuak dira interesgarria zer den edo irudian nabarmentzen dena definitzen dutenak. Irudien biraketa, uzkurdura, translazioa, distortsioa eta abarrekiko aldaezinak dira.

3. *"Orientation assignment "*, gako-puntuaren kokapen bakoitzari orientazio bat edo gehiago esleitzen zaizkio, tokiko irudi-gradientearen norabideetan oinarrituz. Etorkizuneko eragiketa guztiak, ezaugarri bakoitzari esleitutako orientazio, eskala eta kokapenarekin alderatuta eraldatu diren irudi-datuekin egiten dira, eraldatze horiei inbariantzia emanez [21].



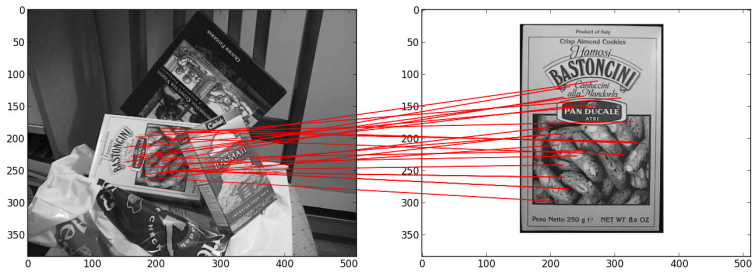
2.6 Irudia: Orientazio asignazioa [3]

4. *"Keypoint descriptor"*, Tokiko irudi-gradienteak gako-puntu bakoitzaren inguruko eskualdean hautatutako eskalan neurtzen dira. Hauek tokiko formaren distortsio-maila nabarmenak eta argiztapen-aldaketak ahalbidetzen dituen irudikapen batean bihurtzen dira [21].



2.7 Irudia: Gako-puntuen deskribatzaileak [3] funtsezko puntuak alderatzeko modua dira. Bektorial formatuan (luzera konstantekoak) laburbiltzen dituzte gako-puntuei buruzko zenbait ezaugarri. Adibidez, haien intentsitatea izan daiteke orientazio nabarmenenaren norabidean. Zenbakizko deskribapena esleitzen dio gako-puntuak aipatzen duen irudiaren eremuari.

5. *"Keypoint matching"*, bi irudien arteko gako-puntuak parekatzen dira hurbilen dauden bizilagunak identifikatuz. Baina kasu batzuetan, bigarren bat etortze hurbilena lehenengotik oso gertu egon daiteke. Zaratagatik edo beste arrazoi batzuengatik gerta daiteke. Kasu horretan, distantzia hurbilenaren eta bigarren distantziaren arteko erlazioa hartzen da [21].



2.8 Irudia: Gako-puntuen bat etortzea, hau da, aurretik kalkulatu diren gako-puntuak eta bere deskriptoreak erabiliz bi irudietan berdinak diren puntuak lotzea.

Ikuspegi honi eskala aldaezinaren ezaugarrien eraldaketa, ingelesez: *"Scale Invariant Feature Transform"* edo *SIFT* izena jarri zaio, irudien datuak tokiko ezaugarriekiko eskala aldaezinezko koordenatuetan bihurtzen baititu.

2.1.2.2 ORB

ORB *"Oriented FAST and Rotated BRIEF"*, OpenCV laborategietan garatu zuten Ethan Rublee-k, Vincent Rabaud-ek, Kurt Konolige-k eta Gary R. Bradski-k 2011n [24]. ***FAST***[1] gako-puntuen detektagailuaren eta ***BRIEF***[2] deskribatzaileen arteko fusioa da.
Lehenik eta behin FAST ereduak, emandako irudiaren ezaugarriak detektatzeko erabiltzen da. Piramide bat ere erabiltzen du eskala anitzeko ezaugarriak sortzeko baina, ez ditu ezaugarrien orientazioa eta deskribatzaileak kalkulatzen, beraz, hemen *BRIEF* sartzen da horretarako.

Aurretik esan bezala, ORB-ek *BRIEF* deskribatzaileak erabiltzen ditu, baina honek arazoak izaten ditu errotazioarekin. Beraz, ORB-ek egiten duena *BRIEF* biratzea da gako-puntuen orientazioaren arabera, hau da, adabakiaren orientazioa erabiliz, bere biraketa-matrizea aurkitzen da eta *BRIEF* biratzen du.

ORB irudien bat etortzerako eredu eraginkorra da konputazio-kostuan eta *SIFT eta SURF*-en ordezko doako bezala sortua izan zen, bi hauek patentatuak baitzeuden [26].

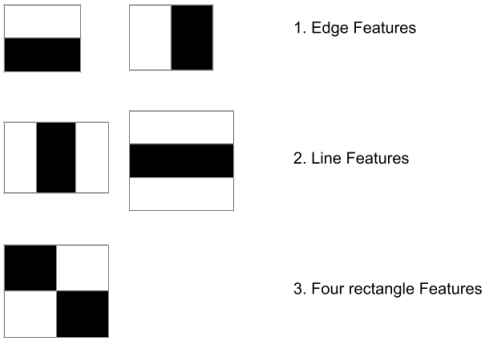
2.2 Ikasketa Automatikoan Oinarritutako Teknikak

2.2.1 Haar Cascades

Viola and Jones-ek *"Haar Classifiers"* izena duen algoritmoa sortu zuten [28]. Zeinek, edozein objektu detektatzeko balio duen, proiektu honen kasuan etiketak. Haar Cascades detektagailuak, *AdaBoost* izeneko sailkatzailea erabiltzen du eta pixelak erabili ordez Haar motako ezaugarriak [3] erabiltzen ditu algoritmo honek. [29]

2.9. irudian ikusten diren ezaugarriak dira eta ezaugarri hauen kalkuluak egiteko, eskualde bakoitzeko pixelen intentsitateak batu eta batuketen arteko ezberdintasunak kalkulatuz egingo dira. Ezaugarri hauek lortzea zaila izan daiteke, irudi handietan beraz, **irudi integralak** erabiltzen dira. Irudi integralak, esentzialki Haar ezaugarrien kalkuluak azkartzen ditu; pixel bakoitza banan-bana konputatu ordez, azpi-laukizuzenak sortzen ditu eta array-erreferentziak sortzen ditu azpi-laukizuzen horietako bakoitzarentzat.

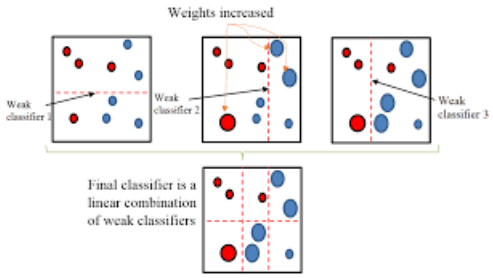
[1] *FAST, Features from Accelerated Segment Test [23]*
[2] *BRIEF, Binary Robust Independent Elementary Features [27]*
[3] Haar motako ezaugarriak, detekzio-leiho batean kokapen zehatz batean aldameneko eskualde laukizuzenetan egiten diren kalkuluak dira, ikus 2.9. irudian ezaugarri mota desberdinak ikusteko.



2.9 Irudia: Haar ezaugarri motak [4], Irudiaren ezaugarri hauek errazten dute irudiko ertzak edo lerroak aurkitzea, edo pixelen intentsitateen bat-bateko aldaketa dagoen eremuak hautatzea.

Garrantzitsua da kontuan izatea Haar-en ezaugarri ia guztiak ez dutela garrantzirik izango objektuak detektatzeko orduan, garrantzitsuak diren ezaugarri bakarrak objektuarenak baitira. Hori lortzeko, zeintzuk diren ezaugarri onenak zehaztu beharko dugu eta horretarako ***AdaBoost*** entrenamendua dugu.

AdaBoost entrenamenduak, sailkatzailea entrenatuko du ezaugarri onenak erabili ditzan. Horretarako, ***"Weak classifiers"*** sailkatzaile desberdinen arteko konbinaketa bat erabiltzen du ***"Strong classifier"*** bat sortzeko, hau da, detekzioetarako indartsuagoa den sailkatzaile bat sortzeko.



2.10 Irudia: Sailkatzaile ahul desberdinen batuketa, indartsuagoa den sailkatzaile bat lortzeko irudia [5].

Azkenik, "cascade"sailkatzailea etapa batzuek osatzen dute, non etapa bakoitza *"weak learner"*-en bilduma den. *"Weak learner"*-ak boosting-a erabiliz trebatzen dira, eta horrek sailkatzaile oso zehatza ahalbidetzen du, *"weak learner"* guztien batez besteko iragarpenetik.

Iragarpen honetan oinarrituz, sailkatzaileak erabakiko du bilatzen dugun objektua aurkitu den (positiboa) edo hurrengo zonaldera mugituko garen (negatiboa). Etapak, adibide negatiboak ahal den azkarren baztertu ahal izateko balio dute. Garrantzitsua da "negatibo faltsu-tasa baxua"maximizatzea, objektu bat ez-objektu gisa sailkatzeak zure objektuak detektatzeko algoritmoa kaltetuko duelako [4].

3. KAPITULUA

Proiektuaren garapena

Jadanik oinarri teorikoak aztertu dira, beraz, proiektuaren garapena azalduko da atal honetan. bi zatitan banatu dena, alde batetik *Eredu Klasikoak* eta bestetik *Eredu Entrenatua.*

3.1 Eredu Klasikoak

3.1.1 Contour Approximation algoritmoa

3.1.1.1 Argazki Transformazioak

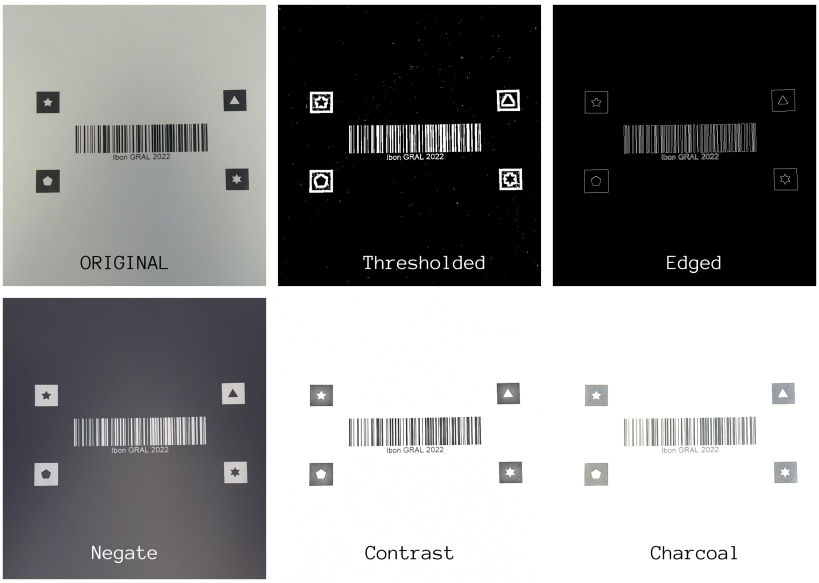
Eredu hau erabili aurretik, argazki datu-base bat behar da, horretarako etiketaren adibide sinple bat sortu da dataset sintetiko bat sortzen hasteko, dibertsitatea izateko honi biraketa aplikatu zaio [1], etiketa gradu desberdinekin izateko; [2]

[1] Biraketak bi eratara egin dira, beltzez eta txuriz beteta biratutako irudiaren atzealdea 3.1
[2] Biraketak irudi originalaren kopiak dira baina zentroarekiko gradu desberdinekin

3.1 Irudia: Biraketa mota desberdinak

Behin argazkiak daudela hauek transformatu egin behar dira, mota askotan egin daitekeena eta beraz, emaitza desberdinak lortu.

Proiektu honetan hurrengo transformazioak aplikatu dira, *Edged*, *Contrast*, *Threshold*, *Charcoal*, *Negate*; 3.2 argazkian ikus daitezke.



3.2 Irudia: Transformazio desberdinak

Behin transformazioa aplikatuta, algoritmoak erabiliko dira argazki horietatik etiketaren *"Bounding Box"*-a[3] atera ahal izateko, hau da, argazki osoa izanik etiketak sortzen duen

[3]Bounding Box-a, euskaraz Muga-koadroa objektuak hautemateko erreferentzia-puntu gisa balio duen

laukizuzena detektatzeko.

3.1.1.2 Detekzioa

Behin transformazioak egin direla detekzioarekin hasteko garaia da. Detekzioarekin hasi aurretik eman beharreko lehen pausua, ertzen detekzioa da. Horretarako *OpenCV* liburutegiak hurrengo funtzioa du, *find_contours*; Funtzio honen bitartez argazkiak dituen ertzak detektatuko dira, ondoren ertz hauek eskuratzeko *imutils* liburutegiaren *grab_contours* funtzioa erabili behar da, ertz horien puntuak lortu ahal izateko; azkenik puntu horiek ordenatu dira, *OpenCV*-ko *sorted* funtzioa erabiliz. Behin ertz-puntu guztiak daudela, hauek iteratuko dira, perimetroa eta puntuaren hurbilketa kalkulatuz, horretarako *arcLength* eta *approxPolyPD* funtzioak eskuragarri ditu hurrenez hurren, *OpenCV*-k.

Behin hurbilketa dugula begiratuko da ea hurbilketa horrek 4 puntu dituen, gure etiketek erreferentzia bezala laukizuzen eta karratuak baitituzte; Hurbilketak 4 puntu izanez gero hau margotua izango da, 3.3 argazkian ikus daitekeen bezala; Azkenik hurbilketa hori bektore batean gordeko da, hurrengo pausuetan *Bounding Box*-a sortu ahal izateko.

3.3 Irudia: Erreferentzia karratuen detekzioa, etiketaren ingurunea sortuko duena

irudimenezko laukizuzena da eta objektu horren talka-koadroa sortzen du. Datu-anotatzaileek laukizuzen hauek irudien gainean marrazten dituzte, irudi bakoitzaren barruan interesgarri den objektua zehaztuz, X eta Y koordenatuak zehaztuz.

3.3 argazkian ikus daiteke nola erreferentziazko 4 laukiak detektatuak daudela, beraz honek detekzioa perfektu burutu dela esan nahi du, bestetik gerta liteke erreferentziazko laukiak detektatzeaz gain barra-kodearen lerro batzuk detektatzea, honek arazoak ekar ditzake bounding box-a marrazterakoan.

3.1.1.3 Bounding Box

Detekzioarekin jarraituz eta behin behar diren puntuak kalkulatuak eta gordeak izan direla, *"Bounding Box"*-a marraztuko da. Horretarako hainbat pausu eta funtzio erabiliko dira, *NumPy* liburutegiaren *array* eta *reshape*, lortutako puntuak array moduan gordetzeko, eta bestetik *OpenCV*-ren *minAreaRect* funtzioa, array-ra bihurtu berriko puntuen area kalkulatzeko, honek behar dugun "kaxa"sortzen baitu.

Behin "kaxa"kalkulatua dagoela, bere ertz puntuak kalkulatuko dira, horretarako "corners"funtzioa sortu dut, zeinek aurretik kalkulatutako "kaxa"parametro bezala izanik, behar ditugun "kaxa-ren lau ertzak lortzen dituen.

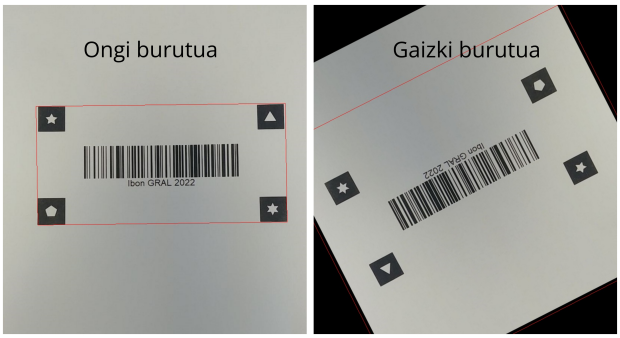
```
cx,cy,w,h,angle = box[0][0],box[0][1],box[1][0],box[1][1],box[2]
CV_PI = 22./7.
_angle = angle*CV_PI/180.;
b = np.cos(_angle)*0.5;
a = np.sin(_angle)*0.5;

pt = []
pt.append((int(cx - a*h - b*w),int(cy + b*h - a*w)));
pt.append((int(cx + a*h - b*w),int(cy - b*h - a*w)));
pt.append((int(2*cx - pt[0][0]),int(2*cy - pt[0][1])));
pt.append((int(2*cx - pt[1][0]),int(2*cy - pt[1][1])));
```

Azkenik, puntu horiek erabiliz *Bounding Box*-a marraztuko da, *OpenCV*-ko *line* funtzioa erabiliz, zeinek marra bat marrazten duen esandako puntu batetik bestera.

3.1.1.4 Warping

Azkenik sortutako *Bounding Box*-a aterako da, *OpenCV* liburutegiaren bi funtzioaz baliatuz. Lehenengo funtzioak aterako den irudiaren perspektiba kalkulatuko du, transfor-



3.4 Irudia: Barra-kodearen Bounding Box mota desberdinak

mazio matrize deritzona, *getPerspective*. Ondoren matrize hau erabiliz eta *warpPerspective* funtzioa erabiliz lortuko da. Egindako probetako emaitzak 4.1.1.

3.5 Irudia: Warping metodoak lortutako irudi posibleak

3.1.2 SIFT eta ORB algoritmoak

Bi algoritmoen prozesua berdina da, aldatzen diren atalak erabilitako ereduak *"SIFT"eta "ORB"* eta eredu hauek erabiltzeko beharrezkoak diren funtzioen deiak dira.

3.1.2.1 Iturburu argazkia

Lehenik eta behin *"template"* argazki bat behar da, hau da, erreferentziazko argazki bat zeinekin parekatu nahi den argazkiarekiko *"match"*-ak ateratzeko balioko duen.

3.6 Irudia: Erabilitako iturburu argazkia

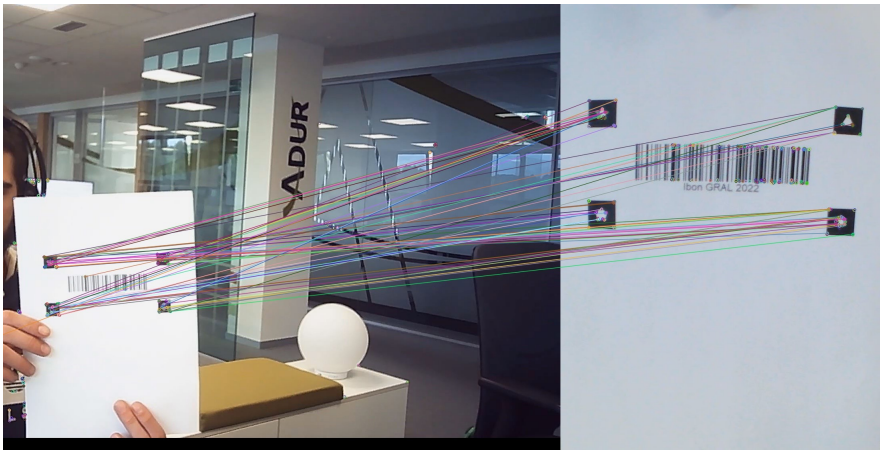
3.1.2.2 Gako-puntuen Bilaketa

Behin erreferentzia jarria, aurretik sortutako dataset-aren gainean algoritmoa aplikatuko da, lehenik eta behin bai erreferentzia irudia bai parekatu nahi den irudia *"Grayscale"*-ra bihurtu dira, hauen *"keypoints"*, hau da, gako-puntuak eta *"descriptors"* deskriptoreak lortzeko, horretarako *detectAndCompute* funtzioari deia egin da.

3.7 Irudia: Parekatu nahi den irudian lortzen diren gako-puntuak, hau da, irudiak dituen interes-puntuak

3.1.2.3 Loturak

Orain puntu guztiak daudela hauen arteko lotura sortu behar da, hau da, *"Match-a*, *Feature Matching*-a egin ahal izateko *OpenCV*-k eskaintzen duen *BruteForceMatcher* funtzioa erabili da, zeinek puntu guztien deskriptoreak begiratu eta hauen arteko loturak egiten dituenak. Puntuak irudian marraztu aurretik hauek ordenatu egin dira soilik hoberenak hartzeko, *"Match"* txarrak saihestu ahal izateko.



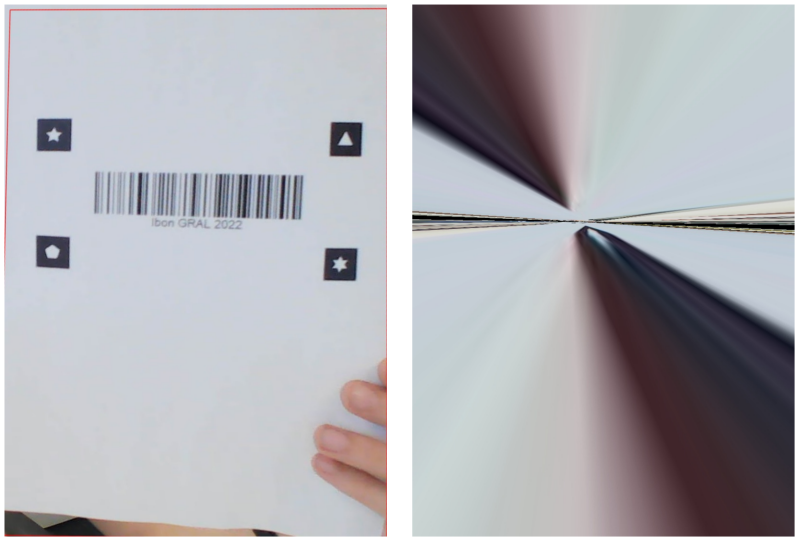
3.8 Irudia: ORB/SIFT algoritmoak lortutako bat etortzeak, argazkian ikusten diren marrak loturak izanik

Loturak marrazteko *OpenCV*-ko *DrawMatches* funtzioa erabili da, honi erreferentziazko

irudia, parekatu nahi den irudia eta azkenik aurreko pausuan lorturiko loturak parametro bezala pasatuz.

3.1.2.4 Parekaketa

Behin lotura guztiak daudela, homografia matrizea bilatu behar da irudia parekatu ahal izateko. *FindHomography* funtzioak prozesu hau beteko du, funtzio honi aurretik lorturiko *keypoint-ak* pasatu behar zaizkio, ondoren matrize hau erabiliz eta erreferentziazko irudiaren altuera eta zabalera erabiliz sortuko dugu irudi parekatua, *warpPrespective* funtzioa erabiliz.



3.9 Irudia: Parekatu nahi den irudian lortzen diren gako-puntuak

3.2 Eredu Entrenatua: Haar Cascades

2.2.1 kapitulua irakurri berriro behar izanez gero. Proiektu hau burutzeko *OpenCV* liburutegiko funtzioak eta Haar Cascades programak erabili dira. [30]
"Weak classifiers"-en kaskada indartsu edo *"Strong classifier"* bat entrenatzeko lagin[4] positibo multzo bat behar da (detektatu nahi diren benetako objektuak dituztenak) eta irudi negatiboen multzo bat (detektatu nahi ez den guztia dutenak, normalean atzealdeak erabiltzen dira, hau da, eredua erabiliko den lekuaren atzealdea detektatu nahi den objektua ez dagoenean). Lagin negatiboen multzoa eskuz prestatu behar da, ondoren lagingailu baten laguntzarekin irudi multzo hauek era bektorialean gorde. Ondoren *"opencv_annotations"* erabiliz, dataset positiboaren gaineko oharkizunak egingo dira, hau da, detektatu nahi den objektua nun dagoen bilatu. [5] Positiboen multzoa sortzeko hainbat era daude:[6]

1. Irudi bakarretik lagin positiboen multzoa sortu, *"opencv_createsamples"* lagingailua erabiliz.

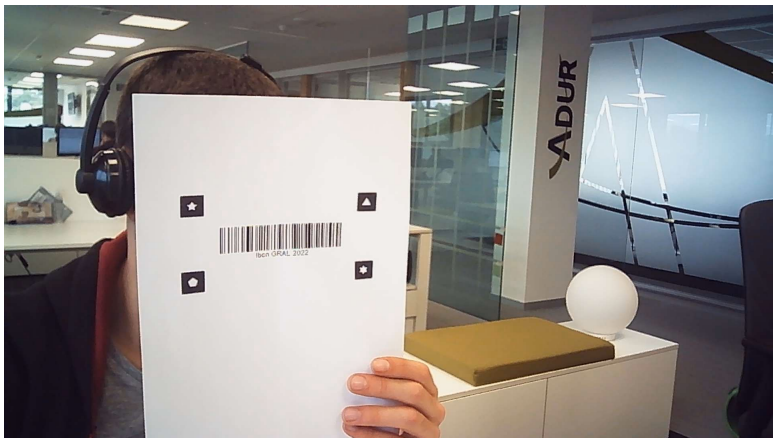
2. Irudi dataset guztia eskuz sortu eta ondoren *"opencv_createsamples"* lagingailua erabili lagin positiboak *OpenCV*-ko formatura egokitzeko.

3.2.1 Dataset-a prestatzen

Aurretik esan bezala, bi lagin multzo behar dira; *positiboen multzoa eta negatiboen multzoa*, hauek eskuz sortzea lan handia da beraz, gauzak erraztearren programa bat sortu da bideo batetik automatikoki argazki dataset-a sortzeko. Programa honek bideo bat hartuko du eta $10 fps/s$-ko frekuentziarekin argazkiak erauziko dira, hau da, 30 segundoko bideo batean 300 argazki erauziko dira. [31]

1. **Positiboen multzoa:** Esan bezala multzo honetan detektatu nahi den objektua azaldu behar da argazkietan eredua entrenatu ahal izateko, horretarako 20 segundoko bideo bat egin da beraz, 200 argazki sortu dira.

[4] Laginak irudietatik ateratzen dira, honekin argazki eredu gehiago sortzea lortzen da
[5] Dataset guztiak, enpresa barruan eta ordenagailuaren web-kamera erabiliz egin dira
[6] Proiektu honetan 200 argazkien dataset-a erabili da lagin positiboak sortzeko



3.10 Irudia: Dataset positiboaren irudi bat

2. **Negatiboen multzoa:** Bestetik negatiboen multzorako, test-a egingo den lekuaren atzealdearen argazkiak beharko dira, ereduari irakasteko zer ez den detektatu behar dena, horretarako berriro ere 20 segundoko bideo bat egin da beraz, 200 argazki sortu dira.

3.11 Irudia: Dataset negatiboaren irudi bat

Positiboen 200 argazkiekin ereduak edo laginak sortu behar dira horretarako *opencv_createsamples* aplikatu da, honi hainbat parametro pasatuz:

1. **Detekzio leihoaren tamaina:** $(width, height)$

2. **Sortu nahi den lagin positiboen kopurua:** Argazki kopurua $\leq$ Lagin kopurua

3.2.2 Irudi positiboen oharkizunak

Behin, irudi positiboen bektore fitxategia eta horretaz gain negatibo eta positiboen dataset-ak eginik daudela, irudi positiboen gainean oharkizunak egin beharko dira. Horretarako, *"opencv_annotations"* aplikazioa erabiliko da, zeinek positiboen dataset-eko irudi guztiak banan banan korrituko ditu, hauetan eskuz markatu ahal izateko detektatu nahi den objektua non dagoen. Marka hauek, objektuaren gainean "bounding-box"bat marraztea izango da, ondoren entrenatzerako garaian ereduak jakin dezan zer den ikasi behar duena.

Hau egin ostean ".txt"fitxategi bat sortuko da zeinek, irudi positiboen izena, detektatu nahi diren objektu kantitatea eta hauen koordenatuak irudian, gordeko dituen. [31]

3.2.3 Ereduaren Entrenamendua

Azkenik ereduaren entrenamendua egin behar da, horretarako *"opencv_traincascade"* aplikazioa erabili da, aurretik lortutako fitxategiak eta parametroak input bezala erabiliz.

1. Erabiliko diren argazkien lagin kantitateak, positiboak eta negatiboak; ohikoenak dataset-aren argazki kopurua erabiltzea edo argazki kopuruaren bikoitza erabiltzea.
2. Argazki positiboen bektore fitxategia, argazkiak bektorizatu eta geroko fitxategia.
3. Argazki positiboen oharkizun fitxategia, aurretik eskuz markatutako ".txt"fitxategia.
4. Egin nahi diren etapa kopurua.
5. Detekzio leihoaren tamaina, detektatu nahi den objektuaren tamaina minimoa.

Proiektu hau burutu ahal izateko hainbat entrenamendu egin dira eredu apropos bat bilatu den arte. Hauetan argazki dataset berdina erabili da. Entrenamendu hau bukatzerakoan "cascade.xml"fitxategia sortua izango da, eta honek eredua gordeko du. [31]

3.2.3.1 **Lehen entrenamendua**

Lehenik eta behin lehen entrenamendua egin aurretik, lagin positiboen bektore fitxategia behar da horretarako lagin kopurua 1000-ra ezarri da, hau da, 1000 lagin positibo sortuko dira. Ondoren entrenatzeko erabiliko direnak, parametroetan esango da, hau da, lagin guztietatik zenbat izango diren erabiliak (ausaz). Lehenengo entrenamenduaren parametroak hurrengoak izan dira:

Etapa Kopurua	Laginen Kopurua	Denbora(minutu)	Emaitza adibidea
10	400	35	3.12

3.1 Taula: Lehenengo entrenamenduaren parametroak

3.2.3.2 Bigarren entrenamendua

Lehenengo entrenamenduko eredutik ondorioztatu da, positiboen lagin kopurua oso baxua ezarri dela beraz, hau aldatu 10000 lagin sortzeko eta berriro entrenatu da eredua. Erabiltzen diren lagin kopuruak, berriro ere argazki kopuruaren bikoitza aukeratu da, gomendagarria baita kopuru horrekin lan egitea.[7] Bestetik aurreko entrenamenduak baino 5 etapa gehiago izango ditu. Bigarren entrenamenduaren parametroak hurrengoak izan dira:

Etapa Kopurua	Laginen Kopurua	Denbora(minutu)	Emaitza adibidea
15	400	45	3.13

3.2 Taula: Bigarren entrenamenduaren parametroak

3.2.3.3 Hirugarren entrenamendua

Aurreko kasuaren alternatiba da entrenamendu hau, sortutako lagin kopurua mantendu da, hau da, 10000 lagin. Bestetik sortutako lagin hauetatik zenbat erabiliko diren aldatu da entrenamendua burutzeko, oraingoan erabaki da laginen kopurua argazki kopuruaren berdina izatea, aurreko kasuarekin emaitzak konparatu ahal izateko. Hirugarren entrenamenduaren parametroak hurrengoak izan dira:

Etapa Kopurua	Laginen Kopurua	Denbora(minutu)	Emaitza adibidea
10	200	27	3.14

3.3 Taula: Hirugarren entrenamenduaren parametroak

3.2.3.4 Azken entrenamendua

Azkeneko entrenamenduari dagokionez, aurreko kasuaren berdina dugu etapa kopurua kontuan hartu gabe. Etapa kopurua 15-era handitu da ikusi baita aurreko kasuaren en-

[7] Lagin Kopuruaren kopuru gomendagarriak argazki kopuruarekiko: Bikoitza, erdia, berdina

trenamenduak emaitza oso zehatzak dituela, beraz hori hobetzea pentsatu da. Azkeneko entrenamenduan erabili diren parametroak:

Etapa Kopurua	Laginen Kopurua	Denbora(minutu)	Emaitza adibidea
15	200	38	3.15

3.4 Taula: Laugarren entrenamenduaren parametroak

3.2.4 Ereduaren erabilera

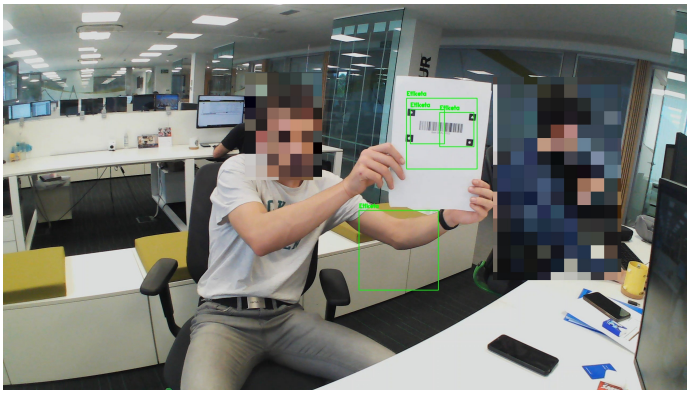
Entrenamendu desberdinetan, parametroak aldatzen dira eta hauen arteko diferentziak edo hobekuntzak ikusi ahal izateko denbora errealean objektua detektatzen duen programa txiki bat egin da, hau da, web-kamera erabiliz momentuan objektua detektatzen duen algoritmoa. Horretarako *"opencv_CascadeClassifier"* funtzioa erabili da zeinek aurreko ataleko ereduak kargatuko dituen eta ondoren eredu hauek erabiliz *"detectMultiScale"* funtzioari deia eginez. [32]

- **Bounding Box kopurua**, hau da, irudiaren gainean aurkitu nahi diren objektu kopurua: 1 objektuarekiko detekzio bakarra nahi baitugu.

- **Detekzio leihoaren tamaina minimoa:** $(48, 48)$ ezarri da tamaina txikiagoko objektua ez baita egongo.

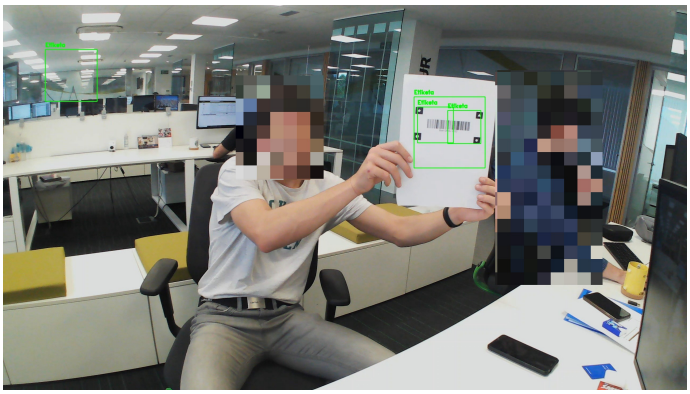
- **Eskala faktorea**, hau da, irudi-eskala bakoitzean irudiaren tamaina zenbat murriztzen den zehazten duen parametroa: 1.02 ez baitugu nahi irudia asko murriztea.

Entrenamendu desberdinen "cascade.xml"fitxategiak erabili dira, hauetan ikus daiteke nola entrenamenduak aldatu ahal detekzioa hobetzen joan den: 3.12 Ikus daiteke detekzioa



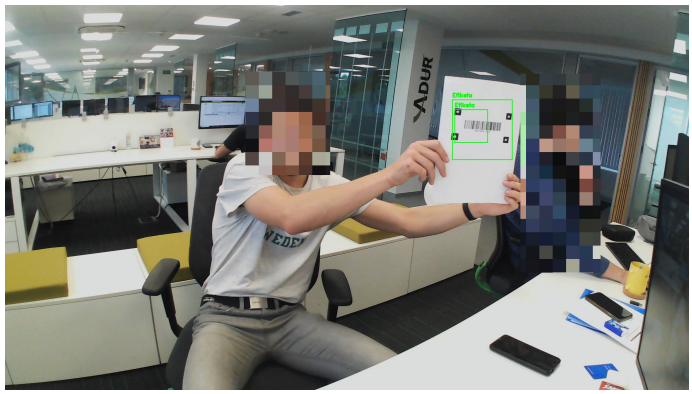
3.12 Irudia: Entrenamendu desberdinen detekzio adibide bat

ez dela gaizki burutzen baina, besoaren zati bat baita ere detektatzen du eta ez dago lauki bakarra, hau da, detekzio bakarra.



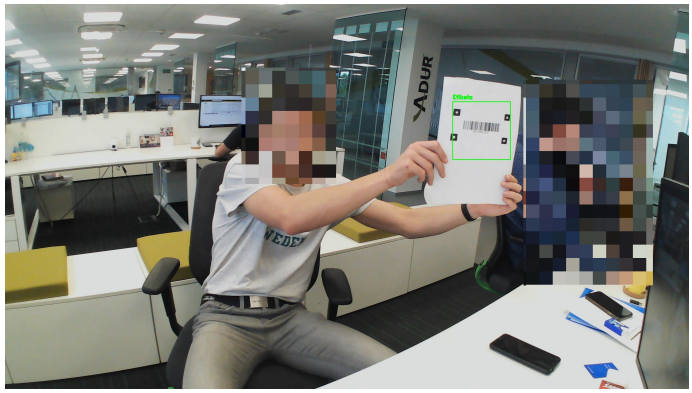
3.13 Irudia: Entrenamendu desberdinen detekzio adibide bat

3.13 Oraingoan eskuaren detekzioa ekidin da baina, etiketa detektatzeaz gain bulegoaren atzealdea detektatzen du.



3.14 Irudia: Entrenamendu desberdinen detekzio adibide bat

3.14 Entrenamendu honetan detekzioa nahiko zehatza da, lortu da ekiditea eskuaren detekzioa bai bulegoaren atzealdearen detekzioa. Soilik etiketa detektatzen da baina, oraindik ez da egin daitekeen detekzio zehatzena, hainbat bounding box baititu.



3.15 Irudia: Entrenamendu desberdinen detekzio adibide bat

3.15 Azkenik lortu da detekzio bakarra, lauki bakarrarekin egitea, hau da, etiketa detektatzea. Aurreko entrenamenduari etapa gehiago ezartzeak arrakasta ekarri dio ereduari beraz, hau izango da 4 atalean erabiliko den eredua.

4. KAPITULUA

Esperimentazioa

Emaitzak hurrengo erara baloratu dira, alde batetik *detekzioa* burutu bada 1 eta burutu ez bada 0, hau da, irudiaren gainean bounding box-a sortu ezkero 1, bestela 0. Ez da zertan detekzioa ondo burutu behar aldagai hau 1 izateko, detekzio bat egotearekin nahiko izango da. Bestetik *asmatzeak* daude, zeinetan aurretik lortutako detekzio guztien artetik ondo detektatutako irudiak izango diren.

Atal hau 4 azpiataletan banatu da, egin diren proba guztientzako atal bat izanik. Guztira 4 proba desberdin antolatu dira, lehenengo probak dataset sintetiko baten gainean egin dira eta beste hiruak aldiz bideo "erreal"batzuetatik lortutako irudien gainean egin dira, beraz totalean 8 argazki dataset desberdin egongo dira. *360, 360, 348, 207, 612, 308* tamainako dataset-ak.

4.1 Lehen Probak

Egindako proba guztiak argazki datu base berdinaren gainean egin da, argazki hauek 1920x1080-ko dimentsioa izango dute. Aurreko atalean esan bezala, irudi originala 360 gradutara biratu da, bi eratara:

1. Biraketa Beltza, irudi originala biratzean sortzen diren hutsuneak beltzez beteko dira.

2. Biraketa Zuria, irudi originala biratzean sortzen diren hutsuneak zuriz beteko dira.

Honekin 360 argazki desberdineko bi dataset lortuko dira. Ondoren argazki guztiei transformazioa desberdinak aplikatuz *Contour Approximation* algoritmoa probatu da, bestetik transformatu gabeko dataset-a erabiliz *SIFT eta ORB* algoritmoak probatu dira eta azkenik eraldatu gabeko dataset-arekin baita ere *Eredu Entrenatua, Haar Cascades* probatu da.

Behin hau jakinda, hurrengo taulari erreparatuz gero, argi ikus daiteke transformazio bakoitzak izan duen bai detekzio tasa[1] bai asmatze tasa [2], denak 360 argazkiko datu basearekin probatuak izan dira.

4.1.1 Contour Approximation

Transformazioak	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze Tasa	Denbora(s)
Edged	**360**	**360**	**100**	**100**	**326.7**
Threshold	250	220	61.12	88	327.9
Negate	115	103	31.94	89.57	324.3
Charcoal	107	96	29.72	89.72	327.6
Contrast	101	69	28.06	68.32	322.8

4.1 Taula: Contour Approximation asmatzeak biraketa zuria.

Transformazioak	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze Tasa	Denbora(s)
Edged	**360**	**360**	**100**	**100**	**325.6**
Threshold	315	275	87.5	87.3	324.5
Contrast	148	107	41.11	72.3	322.8
Charcoal	115	83	31.94	72.17	327.6
Negate	108	75	30	69.44	324.3

4.2 Taula: Contour Approximation asmatzeak biraketa beltza.

4.1 eta 4.2 tauletan emaitza antzekoak daude, hau da, portaera antzekoa izan dute transformazioek bi biraketa motetan. Ikusten da bi transformazio nagusi daudela *(Edged, Threshold)*, hau da, bi transformazioek emaitza oso onak dituzten bitartean besteak nahiko emaitza kaxkarrak dituztela. Bi hauen artean, *Edged* transformazioak, guztiak ongi de-

[1] **Detekzioak** Detekzio tasa ateratzeko irudi guztien artean detekzioa dutenak bereizi dira, hau da, irudiko edozein koordenatuetan bounding box-en bat dutenak

[2] **Asmatzeak** Asmatze tasa ateratzeko detektatuak izan diren irudi horien artetik, etiketa ondo detektatu dutenak izango dira

tektatu ditu beraz, transformazio eraginkorrena dela esan daiteke. Denborari erreperatuz gero, guztiek antzeko eraginkortasuna dutela ikusi daiteke 320 eta 330 segundo artean.

4.1.2 SIFT eta ORB

Algoritmoak	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze Tasa	Denbora(s)
SIFT	**360**	**354**	**100**	**98.33**	**173.36**
ORB	360	303	100	84.17	106.16

4.3 Taula: SIFT/ORB asmatzeak biraketa zuria.

Algoritmoak	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze Tasa	Denbora(s)
SIFT	**360**	**351**	**100**	**97.5**	**168.87**
ORB	360	311	100	86.4	95.6

4.4 Taula: SIFT/ORB asmatzeak biraketa beltza.

Oraingoan 4.3 eta 4.4 tauletan begiratuz gero, *SIFT eta ORB* algoritmoen emaitzak daude eta *Contour Approximation*-ekin bezala biraketa motak eraginik ez duela izan esan daiteke. Emaitza onak jaso dira detekziori begiratuz gero biek lortu baitute %100-a eta asmatzeei dagokionez biek %80-tik gorako tasa dute. Denborari erreparatuz gero ikus daiteke *ORB* algoritmoa *SIFT* baino azkarragoa dela, baina eraginkortasun handiago honek ez du asmatzeetan laguntzen bi biraketa motetan tasa baxuagoa baitu *ORB*-ek. Baina *Contour Approximation*-en emaitzekin alderatzen bada biak askoz azkarragoak dira.

4.1.3 Haar Cascades

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze Tasa	Denbora(s)
Haar Cascades	56	17	15.56	30.36	20.53

4.5 Taula: Haar Cascades asmatzeak biraketa zuria.

Azkenik, *Eredu Entrenatua*-aren emaitzak espero bezala ez dira izan onak, eredua ez baitago entrenatua dataset sintetikoaren gainean, mundu "errealeko"argazkien gainean entrenatua dago. 3.2 atalean azaldu den bezala, eredua entrenatzeko irudien bi multzo behar dira positiboak eta negatiboak. Negatiboak kamara egongo den lekuaren atzealdea izanik

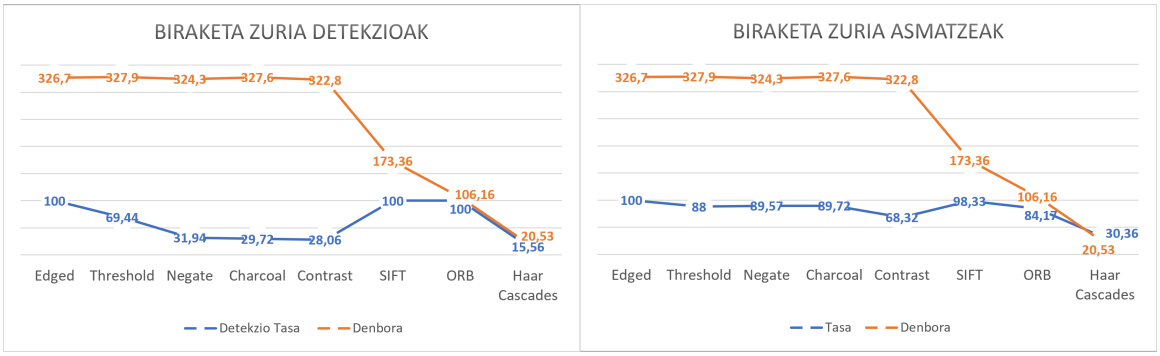
Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze Tasa	Denbora(s)
Haar Cascades	25	9	6.94	36	21.24

4.6 Taula: Haar Cascades asmatzeak biraketa beltza.

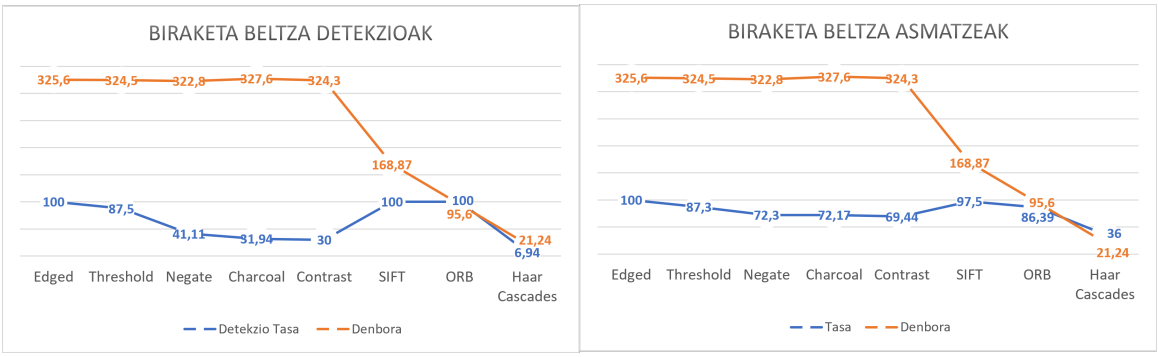
beraz, gure dataset sintetikoan ez dagoenez atzealderik irudi multzo hau ezin da sortu. Hurrengo proba kasuetan ikusiko da eredu entrenatuaren hobekuntza.

4.1.4 Konparaketak

Azpiatal honetan dataset sintetikoaren gainean, algoritmo guztien arteko konparaketa grafikoa dago, hau da, algoritmo guztien bai detekzio bai asmatze tasak eta bakoitzak tasa horiek lortzeko behar izan duen denboraren arteko konparaketa grafikoa. Denbora segundoetan neurtua dago eta aldiz tasa ehunekoetan Argi ikusten da denboraren aldetik



4.1 Irudia: Biraketa Zuriaren Detekzio eta AsmatzeKonparaketa Grafikoa



4.2 Irudia: Biraketa Beltzaren Detekzio eta AsmatzeKonparaketa Grafikoa

eraginkorrenak *HaarCascades, ORB eta SIFT* direla, baina tasari begiratuz gero eraginkorrenak *Contour Approximation:Edged eta SIFT* izan dira, biek %100-eko tasa izanik.

4.2 Bigarren Probak: Errorea

Proba hauetan, behin dataset sintetikoaren emaitzak daudela test errealago batzuk egitea pentsatu da, horretarako bideo motz batzuk grabatu dira *35, 21, 61, 30* segundotako bideoak. Bideo hauen dimentsioa berriro ere $1920x1080$ izanik, hauetatik 10 frame bakoitzetik argazki bat atera da, dataset-a sortzeko. Bideoak 30 frame segundoko frekuentzia izanik hurrengo argazki kantitateak dituzten dataset-ak sortu dira: *348, 207, 612, 308.* 4.2.1 Emaitzak lortzeko soilik *Edge eta Threshold* transformazioak erabili dira, dataset sintetikoari begiratuz gero, guztien artean diferentzia handi batez, emaitza onenak dituztenak baitira.

4.2.1 Contour Approximation

Eredua	**Detekzioak**	**Asmatzeak**	**Detekzio Tasa**	**Asmatze tasa**	**Denbora(s)**
Edged	118	32	33.91	9.19	357.29
Threshold	49	12	14.08	3.45	356.43

4.7 Taula: 348 argazkiko dataset-aren emaitzak.

Eredua	**Detekzioak**	**Asmatzeak**	**Detekzio Tasa**	**Asmatze tasa**	**Denbora(s)**
Edged	97	26	46.86	7.47	258.42
Threshold	33	7	15.94	3.38	263.67

4.8 Taula: 207 argazkiko dataset-aren emaitzak.

Eredua	**Detekzioak**	**Asmatzeak**	**Detekzio Tasa**	**Asmatze tasa**	**Denbora(s)**
Edged	256	57	41.83	9.31	588.43
Threshold	98	29	16.01	4.74	597.56

4.9 Taula: 612 argazkiko dataset-aren emaitzak.

Emaitz logikoak dira, irudi errealagoak erabiltzean hauek atzealdea dute eta beraz ingurune gehiago daude, honek detekzioan arazoak ekartzen ditu. Emaitz hauek ikustean

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
Edged	103	35	33.44	11.36	323.03
Threshold	42	11	13.64	3.57	323.11

4.10 Taula: 308 argazkiko dataset-aren emaitzak.

ondoriozta daiteke *Edged* transformazioak soinu gehiago baztertzen duela, hau da, irudiaren beharrezkoak ez ditugun ertzak hobeto ezabatzen dituela , *threshold*-en aldiz, soinu gehiagoko irudiak sortzen direnez ertz gehiago detektatzen dira eta beraz ez da behar soilik detektatu nahi den objektua detektatuko.

4.2.2 SIFT eta ORB

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
ORB	19	2	0	0	
SIFT	36	6	0	0	394.78

4.11 Taula: 349 argazkiko dataset-aren emaitzak.

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
ORB	11	5	0	0	
SIFT	24	9	0	0	244.16

4.12 Taula: 207 argazkiko dataset-aren emaitzak.

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
ORB	58	16	0	0	
SIFT	73	24	0	0	732.37

4.13 Taula: 612 argazkiko dataset-aren emaitzak.

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
ORB	22	7	0	0	
SIFT	38	16	0	0	378.28

4.14 Taula: 308 argazkiko dataset-aren emaitzak.

Emaitza hauek ikusi eta gero ondorioztatu da bi algoritmo hauetan erroreren bat dagoela, emaitza oso baxuak izateaz gain, detekzio kopuruak ez baitu bat egiten asmatze kopuruarekin. 4.3 atalean ikusten diren emaitzak kodea konpondu ostean sortu diren emaitzak dira.

4.2.3 Haar Cascades

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
Haar Cascades	168	97	0	0	115.92

4.15 Taula: 349 argazkiko dataset-aren emaitzak.

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
Haar Cascades	105	53	0	0	87.3

4.16 Taula: 207 argazkiko dataset-aren emaitzak.

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
Haar Cascades	306	111	0	0	196.19

4.17 Taula: 612 argazkiko dataset-aren emaitzak.

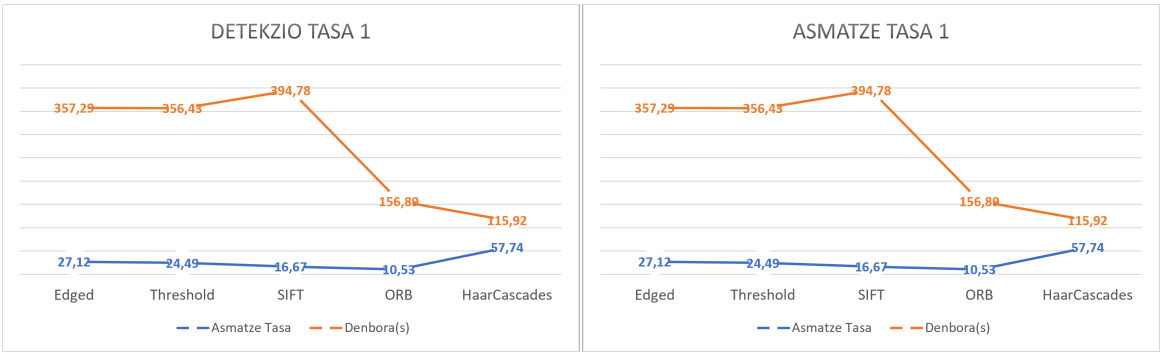
Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
Haar Cascades	136	35	0	0	146.23

4.18 Taula: 308 argazkiko dataset-aren emaitzak.

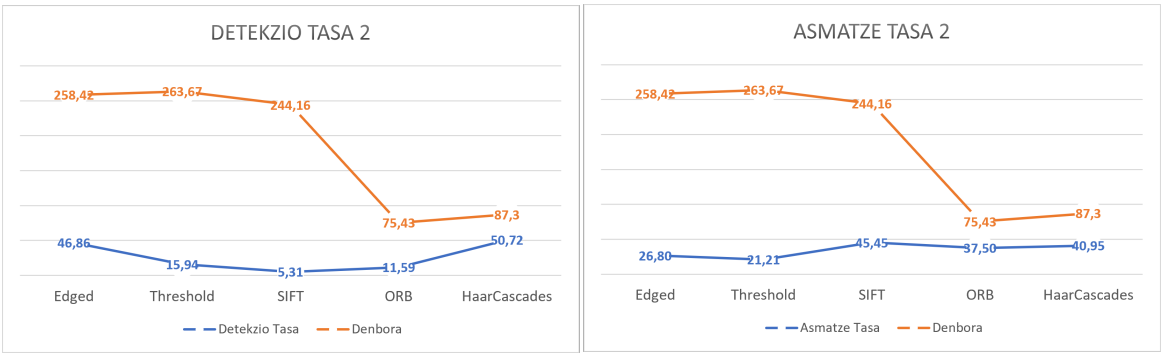
Azkenik aurreko probetan suposatu den bezala, *Haar Cascades* ereduak emaitza hobeak lortu ditu.

4.2.4 Konparaketak

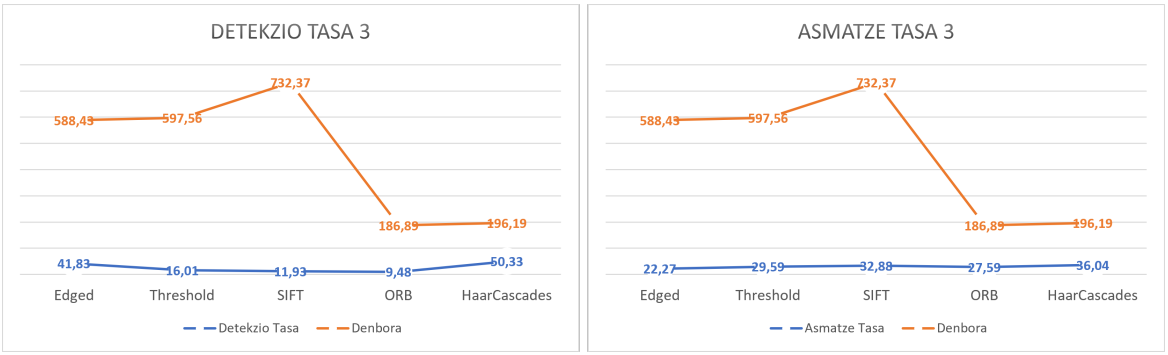
Azpiatal honetan algoritmo guztien arteko konparaketa grafikoa dago oraingoan bideoetatik lortu diren dataset-arenak, hau da, algoritmo guztien bai detekzio bai asmatze tasak eta bakoitzak tasa horiek lortzeko behar izan duen denboraren arteko konparaketa grafikoa. Denbora segundoetan neurtua dago eta aldiz tasa ehunekoetan.



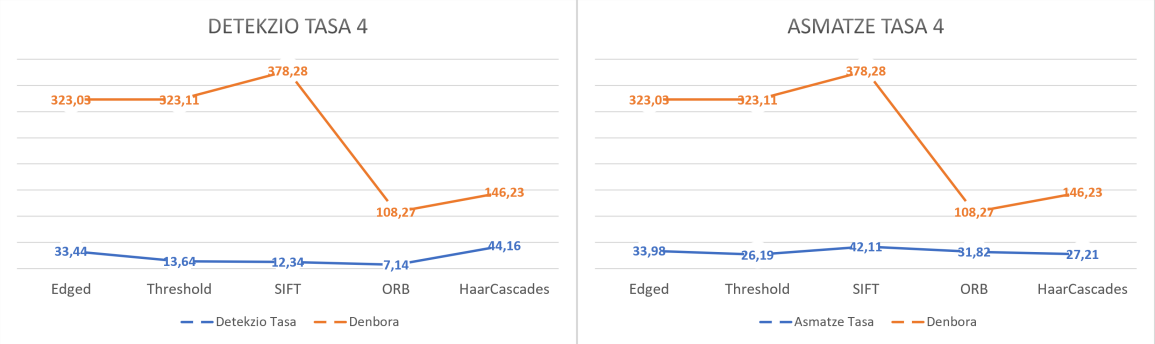
4.3 Irudia: 348 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa



4.4 Irudia: 207 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa



4.5 Irudia: 612 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa



4.6 Irudia: 308 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa

Bideo guztietan bai tasa bai denbora antzekoan exekutatu dira, denborak argazki kopuruarekin proportzionalki lotuak daude, geroz eta irudi gehiago orduan eta denbora gehiago. Argi eta garbi ikusten da SIFT eta ORB algoritmoetan arazoren bat dagoela, ez baitira espero bezain emaitza onak. Besteak aldiz esperotako emaitzen tartean egon dira.

4.3 Hirugarren Probak

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
ORB	140	140	40.87	100	126.89
SIFT	261	261	74.49	100	460.03

4.19 Taula: 349 argazkiko dataset-aren emaitzak.

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
ORB	45	45	21.74	100	85.43
SIFT	113	113	54.59	100	306.1

4.20 Taula: 207 argazkiko dataset-aren emaitzak.

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
ORB	198	198	32.35	100	234.14
SIFT	426	426	69.61	100	1040.87

4.21 Taula: 612 argazkiko dataset-aren emaitzak.

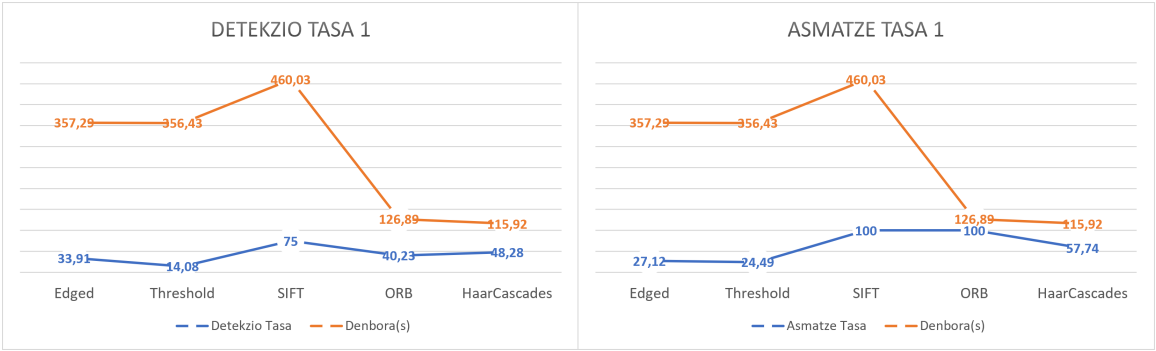
Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
ORB	76	76	24.68	100	120.44
SIFT	157	157	50.97	100	598.51

4.22 Taula: 308 argazkiko dataset-aren emaitzak.

Oraingoan bai ikus daiteke algoritmo hauek emaitza logikoagoak dituztela, hau da, detekzio kopuruak bat egiten duela asmatze kopuruarekin eta horretaz gain detekzio tasa altuagoak daude. Bi algoritmoak konparatuz gero ikusten da *SIFT*-ek emaitza hobeak lortzen dituela baina baita ere denbora gehiago behar duela exekutatzeko.

4.3.1 Konparaketak

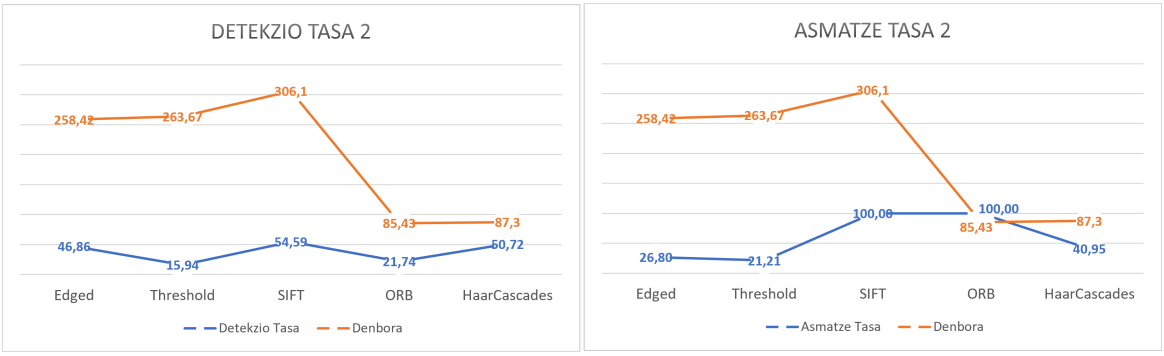
Azpiatal honetan algoritmo guztien arteko konparaketa grafikoa dago oraingoan aurreko atalaren SIFT eta ORB-en emaitzak hobetu ondoren. Algoritmo guztien bai detekzio bai asmatze tasak eta bakoitzak tasa horiek lortzeko behar izan duen denboraren arteko konparaketa grafikoa da. Denbora segundoetan neurtua dago eta aldiz tasa ehunekoetan.



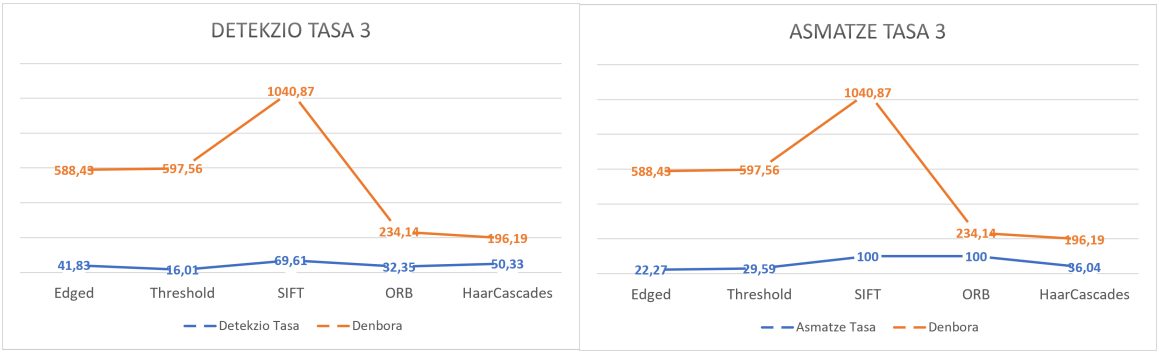
4.7 Irudia: 348 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa

4.4 Laugarren Probak

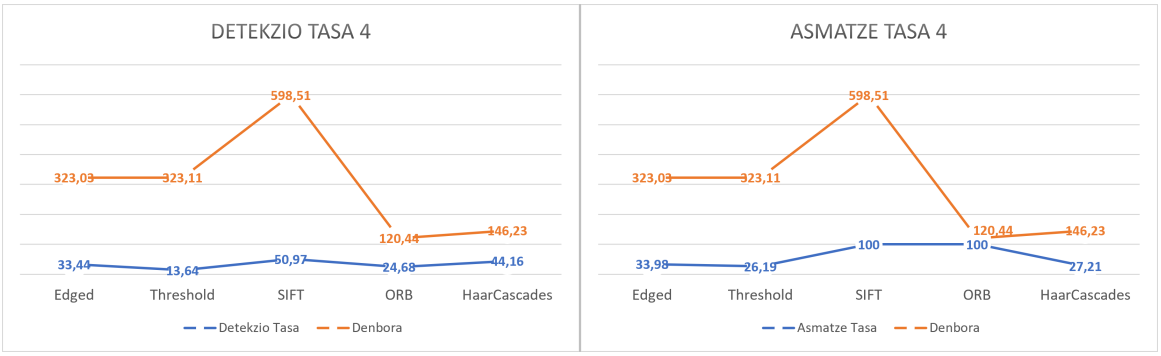
Azken proba hauek burutzeko pentsatu da dataset-eko irudien dimentsioak aldatzea, horretarako kalitate gehiagoko kamara bat erabili da. Dimentsio berria %2560x1440 izanik. Berriro ere 10 frame segunduko frekuentziarekin atera dira irudiak bideoetatik, hurrengo dataset-ak lortuz: *198 argazki, 178 argazki.*



4.8 Irudia: 207 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa



4.9 Irudia: 612 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa



4.10 Irudia: 308 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
Edged	93	32	46.97	16.16	212.30
Threshold	19	7	9.60	3.54	207.55

4.23 Taula: 198 argazkiko dataset-aren emaitzak.

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
Edged	76	24	42.70	13.48	260.52
Threshold	14	5	7.87	2.81	213.71

4.24 Taula: 178 argazkiko dataset-aren emaitzak.

Aurreko bideoen emaitzekin konparatuz gero hobekuntza bat dagoela ikus daiteke, azken finean kalitatea handitzerakoan argazkiek soinu gutxiago dute eta beraz ertzen detekzioak hobeto egin daitezke.

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
ORB	129	129	65.15	100	163.71
SIFT	161	161	81.13	100	534.71

4.25 Taula: 198 argazkiko dataset-aren emaitzak.

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
ORB	129	129	72.48	100	123.37
SIFT	143	143	80.34	100	505.35

4.26 Taula: 178 argazkiko dataset-aren emaitzak.

Berriro ere aurreko kasuekin konparatuz emaitza hobeak lortu dira denbora berdinean.

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
Haar Cascades	138	65	70.41	31.63	56.92

4.27 Taula: 198 argazkiko dataset-aren emaitzak.

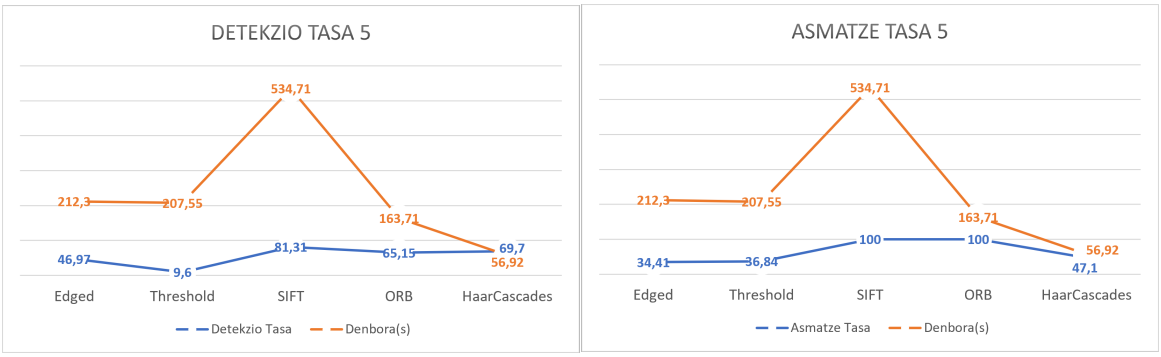
Azkenik, eredu entrenatuak ere emaitzak hobetu ditu, ahala eta guztiz ere ez dira espero bezain onak. Denbora errealean "A simple vista"detekzio hobeak egiten baititu, hau da, kamaren aurrean etiketa mugituz gero denbora errealean detekzioak ongi funtzionatzen du eta etiketa uneoro segitzen baitu.

Eredua	Detekzioak	Asmatzeak	Detekzio Tasa	Asmatze tasa	Denbora(s)
Haar Cascades	125	58	70.23	32.59	52.55

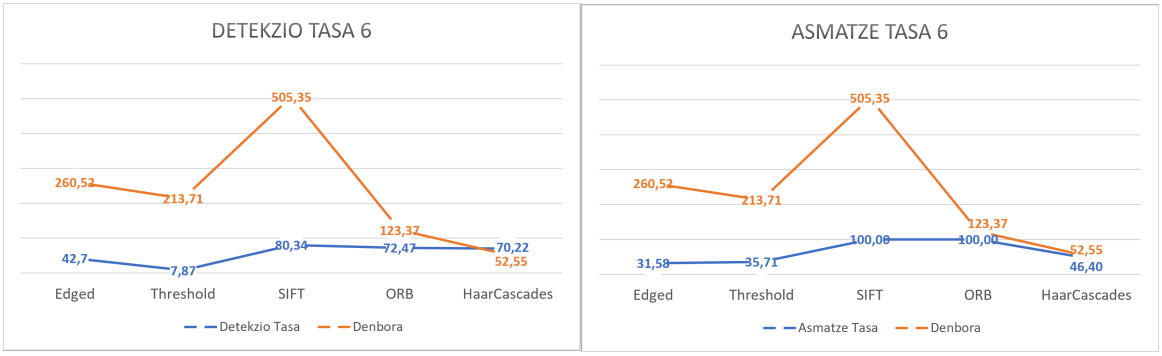
4.28 Taula: 178 argazkiko dataset-aren emaitzak.

4.4.1 Konparaketak

Azpiatal honetan algoritmo guztien arteko konparaketa grafikoa dago oraingoan kamara aldatu eta gero bideoetatik lortu diren dataset-arenak, hau da, algoritmo guztien bai detekzio bai asmatze tasak eta bakoitzak tasa horiek lortzeko behar izan duen denboraren arteko konparaketa grafikoa. Denbora segundoetan neurtua dago eta aldiz tasa ehunekoetan.



4.11 Irudia: 198 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa



4.12 Irudia: 178 argazkiko datasetaren Detekzio eta AsmatzeKonparaketa Grafikoa

5. KAPITULUA

Ondorioak

Proiektua burutzeko ezarritako helburu batzuk bete direla esan daiteke. Izan ere, lana egiteko jakin beharreko kontzeptu teorikoak barneratu dira, hainbat software eta liburutegi aztertu dira, eta arazo desberdinei aurre eginez proiektua amaitzea lortu da.

- **Proiektuari buruzko ondorioak**
  Proiektua burutzearen ondorioak:
  - Contour Approximation algoritmoaren emaitzak ez dira izan espero bezain onak, irudi errealagoak erabiltzeak ertz asko ekartzen baititu eta honek ertzen detekzioa kaltetzen du era nabarmenean beraz, nahiz eta datu sintetikoen gainean aplikatzeko egokienetakoa izan datu errealak erabiltzen direnean ez da kontuan hartuko den algoritmo bat.
  - Irudien bat etortzeetan aldiz ikaragarri hobetu da espero ziren emaitzak, 4.3, 4.4, 4.5, 4.6 emaitzetan ikusten den bezala. Beraz mundu errealarekin nahiz dataset sintetikoarekin erabili daitekeen algoritmoak dira.
  - Irudien dimentsio aldaketaren eragina, irudien kalitateak eragin handia dute detekzioan. Irudi horietan behar ez den soinua baztertzea oso garrantzitsua izango da, gero eta soinu gutxiago egon orduan eta definizio gehiago izango du, hau da, argazki gardenak oso garrantzitsuak dira.
  - Eredu Entrenatuen "arazoa", eredu entrenatuen arazoa entrenamendurako erabiliko diren datuak sortzea da, hau da, behar izan diren dataset egokien sorrera. Ez da zertan dataset handia egin beharrik, dataset horretan behar den guztia

argi eta garbi azaltzea baizik. Horretarako, dokumentazioan azaltzen ez diren entrenamendu desberdinak probatu izan dira dataset egokia topatu arte, behin dataset egokia topatu dela, entrenamendu asko eginez eredua fintzea lortuko da. Denbora gehiago eskaini ezkero eredu hobeak lortuko lirateke, dataset-ak handitzen doazen heinean entrenamendutik lortutako eredua hobetzen joango baita. Beraz, eredu entrenatuetan argi dago eredua ez badago argazki horien gainean entrenaturik arrakasta ez duela izango, bestetik eredu hauek denbora errealean erabiltzeko aproposagoak dira.

- **Ondorio pertsonalak**

  Proiektu hau enpresa batean garatzeak onura dezente ekarri dizkit: enpresa munduaren ezagutza, enpresaren lan moduaren ezagutza, egindako ikasketa desberdinak, arazoei aurre emateko soltura lortzea, eta abar... Egindako lanari buruz, hau da, detekzioari buruz jakintza txikia nuen beraz, lana egiteko informazioa bilatuz, irakurriz, barneratuz eta erabiliz oinarri teoriko asko ikasi ditut. Horretaz gain proiektuan zehar erabilitako tresna eta liburutegiak (OpenCV, imutils ...) erabiltzen ikasi dut. Honi esker etorkizunera begira erraztasunak ekarriko dizkit ikusmen artifizialeko proiektuak garatzeko.

  Orokorrean esan nezake asko ikasi dudala bai enpresaren munduari buruz bai ikusmen artifizialaren munduari buruz, honi esker enpresa mailan proiektuak egiten jarraitzeko gogoa piztu dit. Lan egiteko era polita da, proiektu bat hasieratik ateratzen edo hobetzen joatea, hau da, produktu bat sortzen saiatzea eta ez bakarrik ikerketa egitea.

- **Etorkizunerako Lana**

  Lehenik eta behin zer esanik ez lortutako eredu entrenatua hobetu daiteke, geroz eta argazki datu base handiagoa lortuz eta eredua gehiago entrenatuz. Honi denbora gehiago eskaini ezkero eredu askoz zehatzagoak egongo dira, bestetik eredu desberdinak sor daitezke, hau da, dataset-a guztiz aldatu ikusteko ea eraginkorragoa den dataset berria.

  Ondoren proiektu honen jarraipen bezala, etiketaren detekzioaren ondoren, hauen edukiaren identifikazioa etor liteke. Honi esker mundu errealari begira Adurrek aplikazio erabilgarri asko sortzeko aukera izango du, bezero desberdinentzako.

Eranskinak

A. ERANSKINA

Proiektuaren Antolakuntza

Proiektuaren atal desberdinei eskainitako denbora orduetan ikus daiteke hurrengo taulan.

Lan-Paketea	Denbora (orduak)
Antolakuntza	**15**
Plangintza	5
Jarraipen Kontrola	10
Ikerketa Prozesua	**120**
Informazio Bilketa	60
Informazio Ikasketa	60
Garapena	**250**
Ikusmen Artifizialeko Tekniken Garapena	170
Tekniken Erabilera	60
Emaitzen Ebaluazioa	20
Dokumentazioa	**100**
Memoriaren Idazketa	85
Proiektuaren Defentsa	15
TOTALA	**485**

A.1 Irudia: Proiektuaren pakete desberdinetan egindako denborak

Azkenik *GANTT* diagrama dugu, hau da, proiektuaren kronograma. Bertan ikus daiteke proiektuaren atal desberdinak burutzeko behar izan diren egunak/hilabeteak eta bakoitzaren hasiera eta amaitze data.

Lan-Paketea		Hasiera	Bukaera	Urtarrila	Otsaila	Martxoa	Apirila	Maiatza	Ekaina	Uztaila
Antolakuntza	Plangintza	2022/01/16	2022/01/30							
	Jarraipena	2022/01/16	2022/06/26							
Ikerketa Prozesua	Bilketa	2022/01/16	2022/02/20							
	Ikasketa	2022/02/20	2022/03/30							
Garapena	Garapena	2022/03/25	2022/06/02							
	Erabilera	2022/05/28	2022/06/05							
	Ebaluazioa	2022/06/05	2022/06/13							
Dokumentazioa	Memoria	2022/04/20	2022/06/26							
	Defentsa	2022/06/26	2022/07/13							

A.2 Irudia: GANTT diagrama

Bibliografia

[1] Li-Hua Gong, Cheng Tian, Wei-Ping Zou, and Nanrun Zhou. Robust and imperceptible watermarking scheme based on canny edge detection and svd in the contourlet domain. *Multimedia Tools and Applications*, 80:1–23, 01 2021.

[2] G LoweDavid. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

[3] Deepanshu Tyagi. Introduction to SIFT(Scale Invariant Feature Transform). *Medium*, 2019.

[4] Aditya Mittal. Haar Cascades, Explained. *Medium*, 2020.

[5] Nabil Zerrouki, Fouzi Harrou, Ying Sun, and Amrane Houacine. Vision-based human action classification using adaptive boosting algorithm. *IEEE Sensors Journal*, 18:5115–5121, 2018.

[6] OpenCV team. Opencv.

[7] NumPy team. Numpy.

[8] Matplotlib team. Matplotlib.

[9] Adrian Rosebrock. *PyImageSearch Gurus Course*.

[10] Xin-Yi Gong, Hu Su, De Xu, Zhengtao Zhang, Fei Shen, and Hua-Bin Yang. An overview of contour detection approaches. *International Journal of Automation and Computing*, 15:1–17, 06 2018.

[11] Shay Zweig and Lior Wolf. Interponet, a brain inspired neural network for optical flow dense interpolation, 2016.

[12] Sevim Erdede and Sebahattin Bektas. Examining the interpolation methods used in forming the digital elevation models. *Research Gate*, 06 2020.

[13] Amit Satish Unde, V. A. Premprakash, and Praveen Sankaran. A novel edge detection approach on active contour for tumor segmentation. In *2012 Students Conference on Engineering and Systems*, pages 1–6, 2012.

[14] Fari Abubakar. A study of region-based and contourbased image segmentation. *Signal & Image Processing : An International Journal*, 3:15–22, 12 2012.

[15] Paul Aljabar and Mark J. Gooding. The cutting edge : Delineating contours with deep learning. In *The cutting edge*, 2017.

[16] Deepanshu Tyagi. Introduction To Feature Detection And Matching. *Medium*, 2019.

[17] Edgar Steven Correa-Pinzón. Recognition of objects with feature matching and ransac algorithm. Technical Report ISSN-e 2248-4728, ISSN 1909-9746, Universidad de La Salle, Localización electrónica, 2019.

[18] Natividad Grandón-Pastén, Diego Aracena-Pizarro, and Clésio Tozzi. ReconstrucciÓn de objeto 3d a partir de imÁgenes calibradas 3d object reconstruction with calibrated images. *Ingeniare : Revista Chilena de Ingeniería*, 15, 08 2007.

[19] Samuel Pardo Alia. Detección y seguimiento de objetos mediante precisión-tracking. Master's thesis, Universidad de Alcalá. Escuela Politécnica Superior, 2018.

[20] OpenCV. Harris corner detection. https://docs.opencv.org/4.x/dc/d0d/tutorial_py_features_harris.html.

[21] OpenCV. Introduction to sift (scale-invariant feature transform). https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html.

[22] Minghao Ning. Introduction to SURF (Speeded-Up Robust Features). *Medium*, 2019.

[23] Luca Baroffio, Matteo Cesana, Alessandro Redondi, and Marco Tagliasacchi. Fast keypoint detection in video sequences, 03 2015.

[24] OpenCV. Orb: Oriented fast and rotated brief. https://docs.opencv.org/3.4/d1/d89/tutorial_py_orb.html.

[25] Yanli Liu, Zhang Heng, Hanlei Guo, and Neal.Ñ Xiong. A fast-brisk feature detector with depth information. *National Library of Medicine*, 2018.

[26] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf, 11 2011.

[27] OpenCV. Brief(binary robust independent elementary features). https://docs.opencv.org/3.4/dc/d7d/tutorial_py_brief.html.

[28] Mrinal Tyagi. Viola Jones Algorithm and Haar Cascade Classifier. *Medium*, 2021.

[29] Wikipedia. Cascading Classifiers. *Wikipedia*, 2022.

[30] Ben. Opencv object detection in games. https://learncodebygaming.com/blog/training-a-cascade-classifier.

[31] GeeksForGeeks. Python | haar cascades for object detection. https://www.geeksforgeeks.org/python-haar-cascades-for-object-detection/.

[32] Gabriela Solano. Como crear tu propio detector de objetos con haar cascade | python y opencv. *omes*, 2019.