

Informatika Ingeniaritzako Gradua

Konputazioa

Gradu Amaierako Lana

Bideoen ekintzen sailkapena datu meatzaritzaren teknikak erabiliz

Egilea

Unai Muñoz Aseguinolaza

Zuzendaria(k)

Basilio Sierra Araujo, Naiara Aginako Bengoa

Gaien aurkibidea

Gaien aurkibidea	2
Irudien aurkibidea	5
Taulen aurkibidea	7
1 Sarrera	9
2 Helburuak	11
2.1 Irismena	11
2.2 Lanaren deskonposaketa	12
2.3 Atazak	12
2.4 Kronograma	14
2.5 Dedikazio estimazioa	15
2.6 Dedikazio erreala	16
2.7 Arriskuen plana	17

3	Baliabideak eta softwarea	18
3.1	Datu-basea	19
3.2	CSV formatua	22
3.3	Irudi digitala	23
3.4	FFmpeg	25
3.5	MATLAB	26
3.6	WEKA	27
3.6.1	ARFF formatua	27
3.6.2	ImageFilters	30
3.6.3	Sailkatzaileak	32
3.6.4	Multi-sailkatzaileak	35
4	Garapena	37
4.1	Bideoen aurreprozesaketa	39
4.1.1	Bideoak zatitu	39
4.2	Bideoen analisia	43
4.2.1	Fluxu optikoa kalkulatu	43
4.2.2	Estatistikoak kalkulatu	47
4.2.3	RGB irudia sortu	50
4.3	Bideoen aurreikuspena	52
4.3.1	ARFF-a sortu	52
4.3.2	Iragazkiak aplikatu	54
4.3.3	Sailkapen sinplea	55
4.3.4	Sailkapen konplexua	56
4.3.5	Bozketa	57

5 Emaidzak	59
5.1 Saillkapen sinplea	59
5.2 Saillkapen konplexua	62
5.3 Alderaketa	64
5.4 Bozketa	66
6 Ondorioak eta etorkizuneko lana	67
6.1 Ondorioak	67
6.2 Etorkizuneko lana	68
7 Eranskina: Kodeak	69
Bibliografia	77

Irudien aurkibidea

2.1	LDE diagrama	12
2.2	Kronograma	14
3.1	BOSS datu-basea	19
3.2	Lehenengo kamerako frame-tarteak	20
3.3	Bigarren kamerako frame-tarteak	20
3.4	Hirugarren kamerako frame-tarteak	20
3.5	Laugarren kamerako frame-tarteak	20
3.6	Bosgarren kamerako frame-tarteak	21
3.7	Seigarren kamerako frame-tarteak	21
3.8	Zazpigarren kamerako frame-tarteak	21
3.9	Zortzigarren kamerako frame-tarteak	21
3.10	Bederatzigarren kamerako frame-tarteak	21
3.11	Hamargarren kamerako frame-tarteak	21
3.12	BOSS datu-basearen bideoen aurreikuspena	22
3.13	Gorri, berde eta urdin intentsitateko geruzak	24
3.14	R kanala, G kanala, B kanala eta RGB irudia	24
3.15	FFmeg icon	25
3.16	MATLAB icon	26

3.17	MATLAB interface	26
3.18	WEKA icon	27
3.19	WEKA aplikazioaren interfaze grafikoak	28
3.20	ImageFilters zerrenda WEKA-n	31
3.21	Sailkatzaile motak WEKA-n	32
3.22	K-NN, K=3 izanik	33
3.23	Naive Bayes eredu probabilitistikoa	33
3.24	Bagging eta boosting	35
3.25	Stacking	36
4.1	Garapena	38
4.2	New dataset	42
4.3	SITTING klaseko bideoen aurreikuspena	42
4.4	Horn–Schunck fluxu optikoaren ekuazioa	44
4.5	Fluxu optikoaren adibidea	44
4.6	Optical flow	45
4.7	Optical flow CSV dataset	46
4.8	Parametro estatistikoaren grafikoak	48
4.9	RGB CSV dataset	49
4.10	R, G, B kanalak eta RGB irudia	51

Taulen aurkibidea

2.1	Dedikazioaren estimazioa	15
2.2	Dedikazio erreala	16
3.1	Taula egitura	23
3.2	CSV egitura	23
4.1	Parametro estatistikoaren adibideak	48
4.2	"sit_cps_c01_v01.avi" bideoaren 9 CSV-ak	49
4.3	"sit_cps_c01_v01.avi" bideoaren Vx aldagaiaren geruzak	50
4.4	"sit_cps_c01_v01.avi" bideoaren Vy aldagaiaren geruzak	50
4.5	"sit_cps_c01_v01.avi" bideoaren mg aldagaiaren geruzak	50
4.6	"sit_cps_c01_v01.avi" bideoaren Vx aldagaiaren irudiak	51
4.7	"sit_cps_c01_v01.avi" bideoaren Vy aldagaiaren irudiak	51
4.8	"sit_cps_c01_v01.avi" bideoaren mg aldagaiaren irudiak	51
4.9	ARFF adibidea	53
4.10	AutoColor.arff emaitzarik altuenak	57
4.11	Binary.arff emaitzarik altuenak	57
4.12	ColorLayout.arff emaitzarik altuenak	57
4.13	PHOG.arff emaitzarik altuenak	58
4.14	Bozketa adibidea	58

5.1	Sailkapen sinplearen emaitzak	61
5.2	Sailkapen konplexuaren emaitzak	63
5.3	Sailkapen sinplearen eta konplexuaren arteko desberdintasuna	65
5.4	Sailkapen sinplearen bozketa	66
5.5	Sailkapen konplexuaren bozketa	66

1. KAPITULUA

Sarrera

Bideo ekintzen sailkapena gaur eguneko alorrik landuenetarikoa da. Gizakiak, teknologiaren garapena dela eta, gero eta makina gehiago erabili nahi ditu lana automatizatzeko. Gainera, eremu askotan aplikatu daitekeela ikusi da: auto autonomoetan, denda edo bankuetako zaintza kameretan, aireportuetako makina eta kameretan, eta abar. Helburu nagusia testuinguru desberdinetan gertatzen ari dena aurreikustea da, makinek erabaki propioak hartu ditzaten.

Hala ere, bideo hauen sailkapena hainbat eragileren mende dago: bideoaren kalitatea eta argia, kameraren posizioa eta kokapena, eszenaren tokia, pertsonen ezaugarri fisikoak... Guzti hau kontuan hartuta, ekintzak aurreikusteko egoera errazagoak edo zailagoak aurkituko ditugu. Hori dela eta, ezinezkoa da aurretik definitutako eredu bat erabiltzea, sekuentzia berdineko bi bideo izanik, eragile hauek desberdinak izan daitezkeelako.

Lan honen helburu nagusia auto autonomoak kontrolatzea da, barnean duen kamera erabiliz, errepidean dauden oinezkoen portaera aurreikusteko. Hainbat egoera kudeatzeko erabiliko da, hala nola, autoaren abiadura moteldu pertsona bat errepidea zeharkatzen ari bada, guztiz gelditu norbait lurrera erori bada edo azeleratu planoan inor ez badago. Ez hori bakarrik, baliteke momentu berdinean pertsona bat baino gehiago egotea eta haien artean elkarrekintza sortzea.

Bideoez osatutako datu-basea, BOSS [4] izenekoa, oinarri moduan hartuz, eta hauek fotograma sekuntziak izanik, irudiez osatutako datu-basea eskuratu dugu. Irudi hauen bitartez, binaka hartuz, fluxu optikoa [5] kalkulatu da eta eskuratutako informazio horrekin irudi berriak sortu dira. Amaitzeko, sailkapena egiteko WEKA [6] erabili da. Hasiera batean,

hasierako bideoei hainbat iragazki aplikatu zaizkie ezaugarri nagusiak lortzeko, eta ondoren bi sailkapen mota egin dira: sinplea eta konplexua. Honekin bukatuz, bozketa teknika aplikatu da.

Robotika eta Sistema Autonomoen Ikerketa taldearen (RSAIT) [3] barruan planteatutako eta garatutako proiektua da. Dokumentu honetan aurkezten den lanaren helburuak taldearen helburuekin erabat bateratuta daude. Datu meatzaritzako eta ikasketa automatikako teknikekin lan egiteaz gain, fluxu optikoa erabiltzen da bideoetan ekintzak detektatu ahal izateko.

Hurrengo ataletan proiektu honen informazio guztia aurkitu daiteke: planteatu den diseinua, erabilitako baliabideak, egindako garapena, lortutako emaitzak eta haien ondorioak. Amaitzeko, etorkizuneko lanari begira hainbat puntu zerrendatu dira.

2. KAPITULUA

Helburuak

Atal hau dokumentazio modura erabili da, non proiektuaren diseinuaren planifikazioa ageri den. Alde batetik, irismena azaldu da, proiektuaren zenbait muga zehaztuz. Bestetik, lana egiterakoan sortu daitezkeen atzerapenak saihesteko arrisku plana sortu da. Gainera, LDE diagrama eta kronograma eraiki dira planifikazioa modu argian ikusteko. Honekin batera, lanaren dedikazio estimazioa eta benetako dedikazioa biltzen dituzten taulak sortu dira.

Planifikazio hau jarraitzea oso garrantzitsua izan da modu ordenatuan lan egiteko. Ez hori bakarrik, atal hau erabilgarria izango da atzerapenak ahalik eta gehien ekiditeko.

2.1 Irismena

Proiektuaren helburua bideoetan agertzen diren ekintzak fluxu optikoa erabiliz lantzea izan da. Aurreprozesaketa egiteko MATLAB [7] softwarea erabili da, eta ondoren, analisisa egiteko WEKA softwarea. Azken honek, ikasketa automatikoan, datu meatzaritzan eta Big Datan oinarritzen diren teknikak erabiltzen ditu.

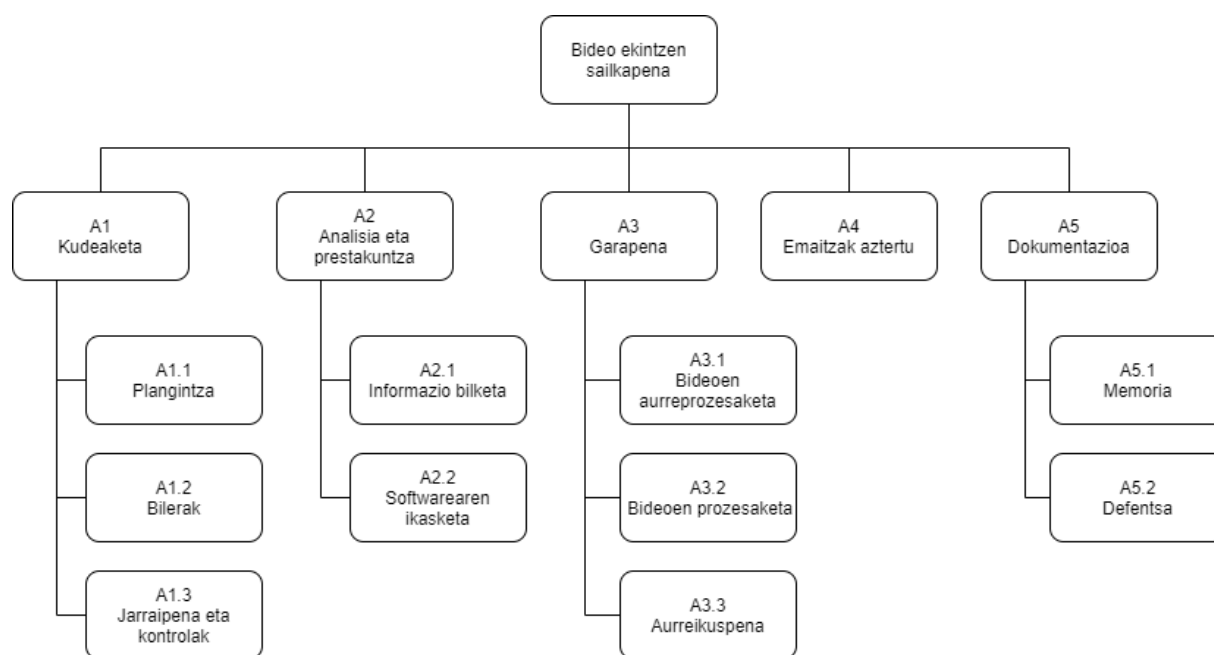
RSAIT ikerketa taldearen barnean lan egin denez, robotika adimentsuan aplikatzeko erabili da proiektua garatzeko. Roboten autonomia areagotzeko helburuarekin, datu analisisiko metodoak ikasketa automatikoekin batera erabili dira.

Denbora kontuan hartuta, proiektua GrAL modura erabiliko denez, unibertsitateak jarri dituen epeetara moldatu behar izan da. Kasu honetan, abenduan hasi zen eta ekainean bukatu da. Aurkezteko epea uztaila da.

2.2 Lanaren deskonposaketa

Lanaren deskonposaketa oso garrantzitsua izan da atal guztiak ondo bereizteko eta planifikazioa modu zuzenean osatzeko. 2.1 Irudiko diagraman proiektu osoko garapena lan-pakete desberdinetan banatu dela ikusi daiteke. Modu berean, lan-pakete bakoitza hainbat atazatan zatitu da.

Guztira bost pakete sortu dira: lehenengo biak, *A1 Kudeaketa* eta *A2 Analisia eta prestakuntza*, lanarekin hasi baino lehen gauzatu dira. *A3 Garapena* eta *A4 Emaizak aztertu* pakete pisutsuenak eta garrantzitsuenak dira. Azkenengoa, *A5 Dokumentazioa*, lana bukatu ondoren gauzatu da eta fitxategi hau pakete honi dagokio.



2.1 Irudia: LDE diagrama

2.3 Atazak

Lanaren deskonposaketa bezala, atal honetan aurrera eraman diren zereregin guztiak zerrendatu dira. Aurreko atalean ez bezala, hemen lan-pakete guztiak eta haien atazak ageri dira. Gainera, ataza batzuk beste ataza desberdinetan banatuak izan dira.

-
- A1. Kudeaketa
 - A1.1 Plangintza
 - A1.2 Bilerak
 - A1.3 Jarraipena eta kontrolak
 - A2. Analisia eta prestakuntza
 - A2.1 Informazio bilketa
 - A2.1.1 Artikuluak
 - A2.1.2 Datu-basea
 - A2.2 Softwarearen ikasketa
 - A2.2.1 Instalazioa
 - A2.2.2 Funtzionamendua
 - A3. Garapena
 - A3.1 Bideoen aurreprozesaketa
 - A3.1.1 Bideoak zatitu
 - A3.2 Bideoen prozesaketa
 - A3.2.1 Fluxu optikoa
 - A3.2.2 Estatitiskoak
 - A3.2.3 RGB irudia
 - A3.3 Aurreikuspena
 - A3.3.1 ARFF prestatu
 - A3.3.2 Sailkapen sinplea
 - A3.3.3 Sailkapen konplexua
 - A3.3.4 Bozketa
 - A4. Emaitzak aztertu
 - A5. Dokumentazioa
 - A5.1 Memoria
 - A5.2 Defentsa

2.4 Kronograma

Proiektua hainbat hilabetetako lana izan denez, aste asko behar izan dira aurreko ataza guztiak burutzeko. 2.2 Irudian proiektuaren plangintza laburbildu da kronograma moduan. Proiektua otsailean hasi zenez, guztira 24 aste behar izan dira proiektua lantzeko eta garatzeko.

Denbora eta dedikazio gehien behar izan duen paketea, A3 *Garapena* lan-paketea izan da, 14 astetan zehar burutu dena: 4. astetik 17. astera. Hori egin ahal izateko, aurretik, lehenengo hiru asteak plangintza, bilerak eta informazio bilketa prestatzeko erabili dira batez ere.

Ataza eta zeregin ugari sortu direnez, zuzendarien feedback-a behar izan da kasu askotan. Hori dela eta, jarraipena eta kontrola (A1.3 lan-paketea) denbora osoan zehar eman da. Ez hori bakarrik, bilerak (A1.2 lan-paketea) 1., 3., 8., 15. eta 17. asteetan egin dira.

Maiatzaren lehenengo astetik aurrera, hau da, 15. astetik, memoria (A5.1 lan-paketea) garatzeko erabili da denbora gehiena. Azkenik, azkenengo hiru asteak defentsa (A5.2 lan-paketea) prestatzeko erabili dira.

	Otsaila					Martxoa					Apirila				Maiatza				Ekaina				
	01	08	15	22	29	01	08	15	22	29	05	12	19	26	03	10	17	24	31	07	14	21	28
A1.1 Plangintza	■																						
A1.2 Bilerak	■		■					■							■		■						
A1.3 Jarraipena eta kontrolak	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
A2.1 Informazio bilketa	■	■																					
A2.2 Softwarearen ikasketa	■	■	■					■				■			■								
A3.1 Bideoen aurreprozesaketa				■	■																		
A3.2 Bideoen prozesaketa						■	■	■	■	■	■	■	■	■									
A3.3 Aurreikuspena															■	■	■	■	■	■	■	■	■
A4. Emaitzak aztertu																							
A5.1 Memoria															■	■	■	■	■	■	■	■	■
A5.2 Defentsa																							■

2.2 Irudia: Kronograma

2.5 Dedikazio estimazioa

Proiektuarekin hasi aurretik, zeregin guztiak garatzeko lan-pakete bakoitzak behar duen denboraren estimazioa kalkulatu da. Honi esker, lan-karga kalkulatzeari lortu da. Datu guzti hauek 2.1 Taulan batu dira.

Proiektu osoaren garapenak 300 ordu iraungo dituela guztira estimatu zen. Lanarekin hasi baino aurreko zereginak, *A1 Kudeaketa* eta *A2 Analisia eta prestakuntza*, 40 eta 50 orduko estimazioa eman zaie, hurrenez hurren. Atalik luzeena *A3 Garapena* izango dela kalkulatu da, 110 orduko estimazioarekin.

Lan-Paketea	Dedikazio estimazioa (ordutan)
A1. Kudeaketa	40
A1.1 Plangintza	10
A1.2 Bilerak	15
A1.3 Jarraipena eta kontrolak	15
A2. Analisia eta prestakuntza	50
A2.1 Informazio bilketa	25
A2.2 Softwarearen ikasketa	25
A3. Garapena	110
A3.1 Bideoen aurreprozesaketa	10
A3.2 Bideoen prozesaketa	50
A3.3 Aurreikuspena	50
A4. Emaidzak aztertu	20
A5. Dokumentazioa	80
A5.1 Memoria	60
A5.2 Defentsa	20
Guztira	300

2.1 Taula: Dedikazioaren estimazioa

2.6 Dedikazio erreala

Oraingo 2.2 Taulan lan-pakete bakoitzak behar izan duen dedikazio erreala biltzen da. Orokorrean nahiko ondo estimatu da orduen banaketa, guztira 12 orduko desbiderapena gertatu baita.

A3 Garapena eta A5 Dokumentazioa lan-paketetan estimatu den baino ordu gehiago erabili dira, 15 ordu bakoitzean. A1 Kudeaketa, A2 Analisia eta prestakuntza eta A4 Emaitzak aztertu lan-paketetan, aldiz, gutxiago.

Lan-Paketea	Dedikazio estimazioa (ordutan)	Desbiderapena (ordutan)
A1. Kudeaketa	35	-5
A1.1 Plangintza	7	-3
A1.2 Bilerak	10	-5
A1.3 Jarraipena eta kontrolak	18	+3
A2. Analisia eta prestakuntza	42	-8
A2.1 Informazio bilketa	22	-3
A2.2 Softwarearen ikasketa	20	-5
A3. Garapena	125	+15
A3.1 Bideoen aurreprozesaketa	10	0
A3.2 Bideoen prozesaketa	60	+10
A3.3 Aurreikuspena	55	+5
A4. Emaitzak aztertu	15	-5
A5. Dokumentazioa	95	+15
A5.1 Memoria	70	+10
A5.2 Defentsa	25	+5
Guztira	312	+12

2.2 Taula: Dedikazio erreala

2.7 Arriskuen plana

Proiektuen garapenean zehar ohikoa da arazoak gertatzea eta ondorioz planifikazioa behar den bezala ez jarraitzea. Hau ekiditeko, arriskuen plana eraiki da, arriskuak bilduz eta haiei irtenbide azkarra aplikatuz.

1. **COVID-19:** Aurten eman den arazorik esanguratsuen COVID-19aren pandemia izan da. Batez ere, zuzendariekin harremanetan jartzeko orduan nabari da. Beste aukerarik egon ez denez, bilera eta jarraipen guztiak online egin dira, bai posta elektronikoz eta Blackboard Collaborate-ren bidez.
2. **Ordenagailuaren matxura:** Proiektuan zehar garatuz joan den lana galtzea ordenagailuaren matxuratzearen ondorioz. Arazo hau ekiditeko, garapenaren aurrerapen guztiak Google Driven gorde dira. Modu berean, aldaketak desegiteko eta aurreko bertsioak berreskuratzeko aukera eduki da.
3. **Atzerapenak:** Ikasleak beste alorretako lan eta ikasketa izan dituzenez, zenbait atzerapen sortu dira. Arazo honen irtenbidea planifikazio zuzena izan da. Astero plangintza eta egin beharreko guztia begiratu da, eta atzerapen bat egonez gero planifikazio berria eratu da momentuko egoeran oinarrituta.
4. **Zailtasunak:** Proiektuaren garapenean hainbat zailtasun aurkitu ditzakegu. Arazo hau ekiditeko, dedikazio handiagoa behar duten atazei denbora gehiago eskainiko zaie. Azkenengo aukera bezala, irtenbidea aurkitu ezean, zuzendariei galdetuko zaie laguntza eske.

3. KAPITULUA

Baliabideak eta softwarea

Atal honetan, proiektua garatzeko erabili diren erremintak eta baliabideak azaltzen dira. Sistema eragilea kontuan hartuta, Linux eta Windows-ekin lan egin da. Ondoren proiektu osoan zehar erabili diren baliabideak eta softwarea zerrendatu da, lehenengoa, datu-basea, oinarri eta abiapuntutzat erabili dela kontuan izanik.

Baliabideak:

- Datu-basea
- CSV formatua
- Irudi digitala

Softwarea:

- FFmpeg
- MATLAB
- WEKA

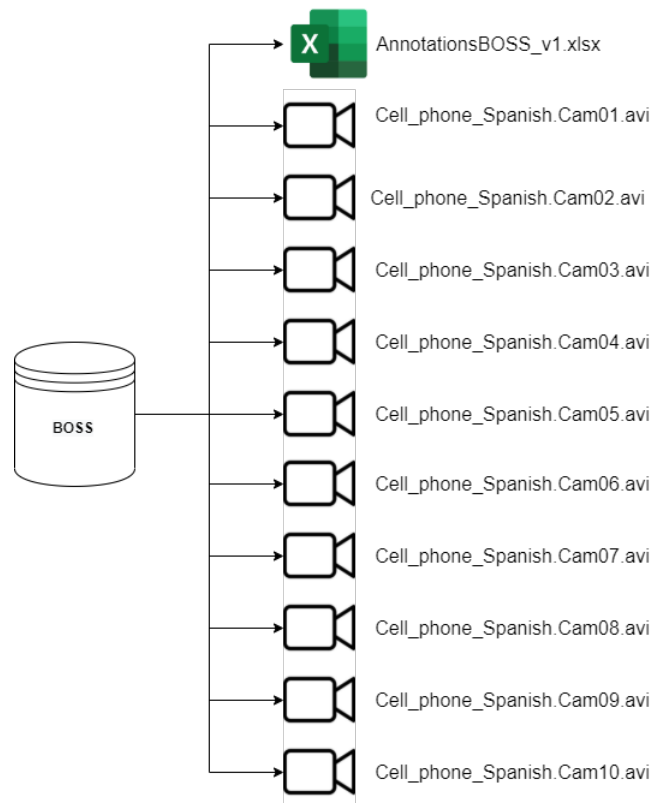
3.1 Datu-basea

Proiektuaren oinarria eta, beraz, baliabiderik garrantzitsuen datu-basea da. Datu-basea **BOSS** izena du eta hainbat bideo multzoz osatuta dago. Multzo bakoitzak, gertakizun bat erakusten du eta bertako bideo bakoitzak horren ikuspegi desberdina.

Erabili den bideo multzoa "Cellphone Spanish" da eta hemendik aurrera gure datu-basea bihurtuko dugu. Datu-basea 10 bideo desberdinez osatuta dago. Bideo guztiek bi pertsonen tren batean duten elkarrekintza berdina erakusten dute eta bakoitza kamera desberdinetik grabatua izan da. Hori dela eta, gertaera berdina, baina ikuspegi desberdinetik biltzen dituzten bideoak ditugu.

Bideo hauetan hiru ekintza desberdin bereizi daitezke: eseri, altxatu eta ibili. Bideoekin batera, *xlsx* fitxategi bat ageri da, ekintza bakoitza bideoen zein momentuan gauzatzen den adierazten duena. Ondorioz, ekintza bakoitzeko klase bat bereiztuko dugu: *sitting*, *standing* eta *walking*.

Datu-basea [3.1](#) Irudiko egitura dauka eta [3.12](#) Irudian bideoen aurreikuspena ageri da.



3.1 Irudia: BOSS datu-basea

"AnnotationsBOSS_v1.xlsx" fitxategiak, pertsona bakoitzak egiten dituen ekintzak bi-deoen zein frame-tartetan gauzatzen diren adierazten du. Adibidez, 3.2 Irudian lehenengo kamerako frame-tarteak ageri dira. Lerro bakoitzean klase desberdineko tartekak ageri dira, eta zutabe bakoitzean hasiera edo bukaera tartekak.

Datu hauen arabera ondorengoa ondorioztatu daiteke: lehenengo pertsona [414,481] eta [1282,1328] frame tartetan ibiltzen ari da. Bigarrena aldiz, [725,835], [990,1057] eta [1091,1129] tartetan. Gainera, lehenengo pertsona [482,1110] tartean eserita dagoela eta [1111,1282] tartean altxatuta dagoela badakigu. Bigarrena, ordea, [836,989] tartean eserita dago eta [1058,1090] tartean altxatuta.

Camera 1											
Cell_phone_Spanish											
Actions	Person 1				Person 2				Start3	Start3	
	Start1	End1	Start2	End2	Start1	End1	Start2	End2			
Walking	414	481	1282	1328	725	835	990	1057	1091	1129	
Sitting	482	1110			836	989					
Standing	1111	1282					1058	1090			

3.2 Irudia: Lehenengo kamerako frame-tarteak

Cell_phone_Spanish											
Actions	Person 1				Person 2				Start3	Start3	
	Start1	End1	Start2	End2	Start1	End1	Start2	End2			
Walking	439	481	1282	1306	762	835	990	1057	1091	1112	
Sitting	482	1110			836	989					
Standing	1111	1282					1058	1090			

3.3 Irudia: Bigarren kamerako frame-tarteak

Cell_phone_Spanish											
Actions	Person 1				Person 2				Start3	Start3	
	Start1	End1	Start2	End2	Start1	End1	Start2	End2			
Walking	459	481	1282	1290	786	835	990	1057	1091	1108	
Sitting	482	1110			836	989					
Standing	1111	1282					1058	1090			

3.4 Irudia: Hirugarren kamerako frame-tarteak

Cell_phone_Spanish											
Actions	Person 1				Person 2				Start3	Start3	
	Start1	End1	Start2	End2	Start1	End1	Start2	End2			
Walking	404	481	1282	1334	706	835	990	1057	1091	1135	
Sitting	482	1110			836	989					
Standing	1111	1282					1058	1090			

3.5 Irudia: Laugarren kamerako frame-tarteak

Cell_phone_Spanish											
Actions	Person 1					Person 2					
	Start1	End1	Start2	End2	Start1	End1	Start2	End2	Start3	Start3	
Walking	424	481	1282	1318	737	835	990	1057	1091	1123	
Sitting	482	1110			836	989					
Standing	1111	1282					1058	1090			

3.6 Irudia: Bosgarren kamerako frame-tarteak

Cell_phone_Spanish											
Actions	Person 1					Person 2					
	Start1	End1	Start2	End2	Start1	End1	Start2	End2	Start3	Start3	
Walking	405	481	1282	1331	706	835	990	1057	1091	1132	
Sitting	482	1110			836	989					
Standing	1111	1282					1058	1090			

3.7 Irudia: Seigarren kamerako frame-tarteak

Cell_phone_Spanish											
Actions	Person 1					Person 2					
	Start1	End1	Start2	End2	Start1	End1	Start2	End2	Start3	Start3	
Walking	362	481	1282	1372	646	835	990	1057	1091	1160	
Sitting	482	1110			836	989					
Standing	1111	1282					1058	1090			

3.8 Irudia: Zazpigarren kamerako frame-tarteak

Cell_phone_Spanish											
Actions	Person 1					Person 2					
	Start1	End1	Start2	End2	Start1	End1	Start2	End2	Start3	Start3	
Walking	394	481	1282	1333	698	835	990	1057	1091	1134	
Sitting	482	1110			836	989					
Standing	1111	1282					1058	1090			

3.9 Irudia: Zortzigarren kamerako frame-tarteak

Cell_phone_Spanish											
Actions	Person 1					Person 2					
	Start1	End1	Start2	End2	Start1	End1	Start2	End2	Start3	Start3	
Walking	369	481	1282	1373	656	835	990	1057	1091	1163	
Sitting	482	1110			836	989					
Standing	1111	1282					1058	1090			

3.10 Irudia: Bederatzigarren kamerako frame-tarteak

Cell_phone_Spanish		
Actions	Person 1	
	Start1	End1
Walking	15	94
Sitting		
Standing		

3.11 Irudia: Hamargarren kamerako frame-tarteak



3.12 Irudia: BOSS datu-basearen bideoen aurreikuspena

3.2 CSV formatua

CSV (comma-separated values) [8] fitxategiak taula itxurako datuak gordetzen dituen formatu irekia da. Normalean informazio guztia puntu eta koma puntuazio-markarekin banatzen da. Izan ere, kalkulu orri baten bidez, hala nola LibreOffice Calc edo Excel bidez, CSV formatuko fitxategi bat irekitzean aplikazioak berak galdetuko digu zein bereizlekin mugatzen ditu datuak. Formatu honetan informazioa biltzean, normalean lehenengo lerroan zutabeen izena gordetzen da.

AMA (American Music Awards) 2015eko urteko irabazleak biltzen dituen 3.1 Taula adibide gisa izanik, 3.2 Taulako forma izango luke CSV formatuan. Lehenengo lerroa aztertuz, atributuen izenak "Maila" eta "Irabazlea" direla badakigu. Modu honetan, puntu eta komaren aurretik datorren informazioa "Maila" atributuari dagokio eta ondorengoa "Irabazlea" atributuari.

Maila	Irabazlea
Gizonezko artista gogokoena	Ed Sheeran
Emakumezko artista gogokoena	Ariana Grande
Urteko abestia	Taylor Swift
Urteko artista	One Direction

3.1 Taula: Taula egitura

Maila;Irabazlea
Gizonezko artista gogokoena;Ed Sheeran
Emakumezko artista gogokoena;Ariana Grande
Urteko abestia;Taylor Swift
Urteko artista;One Direction

3.2 Taula: CSV egitura

3.3 Irudi digitala

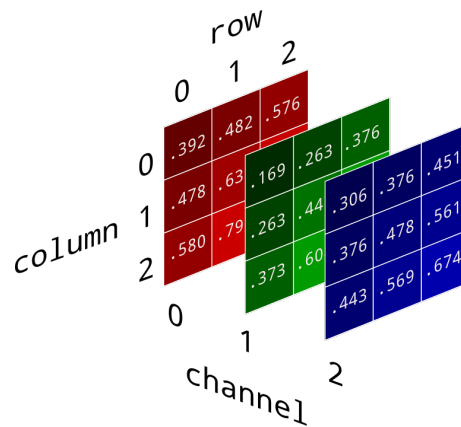
Irudi digitala, matrize-irudia edo bitmap (bit-ez osatutako mapa) bezala ezaguna da. N x M dimentsioko matrizea da, non osagai bakoitzeko zenbaki osoko balioa informazio maila adierazten duen. Hau da, x eta y balioak 0-tik $N-1$ -era eta 0-tik $M-1$ -era mugitzen dira hurrenez hurren. Orduan, $I(x_i, y_j)$ funtzioak (x_i, y_j) pixelaren informazioa gordetzen du.

$$I(x, y) \mid 0 \leq x \leq N - 1 \wedge 0 \leq y \leq M - 1 \mid N, M \in \mathbb{N}$$

Ondorioz, irudi digitala matrize bat da, pixel bakoitzaren informazioa gordetzen duena. Oro har, 8 biteko irudiak erabiltzen dira, beraz pixel bakoitzak $[0, 255]$ bitarteko balioak gordetzen ditu. 0 balioa argi intentsitate txikiena izanik eta 255 altuena.

Demagun matrize bat hainbat matrizez osatuta dagoela, non geruza bakoitzak propietate desberdina duen. Geruza bakoitzari kanala deituko diogu. Adibidez, 3.13 Irudian RGB irudia dugu, non kanal bakoitza Gorri, Berde eta Urdin argi intentsitateekin bat datorren.

Beste modu batera esanda, RGB irudi bat hiru kanalez osatuta dago: gorria (R), berdea (G) eta urdina (B). Lehenengo geruzako pixelek zenbateko gorria duten adierazten dute, bigarren geruzakoek zenbateko berdea eta hirugarren geruzakoek zenbateko urdina. Zuri beltzean dagoen irudi batek, ordea, kanal bakarra izango du.



3.13 Irudia: Gorri, berde eta urdin intentsitateko geruzak

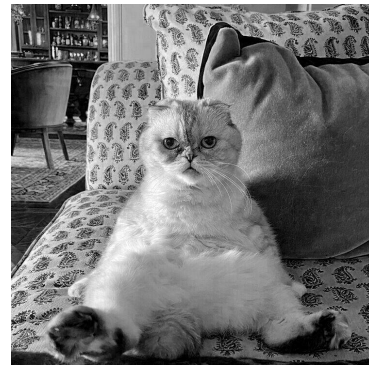
Laburbilduz, irudi bat kanal guztien konbinazioa da. Hau da, irudiaren (X,Y) pixelak, geruza guztien (X,Y) pixelen konbinazioaz sortzen da. 3.13 Irudiko matrizea adibide gisa izanik, (0,0) pixela 392, 169 eta 306 arteko konbinazioa da, kolore zehatz bat sortuz.



3.14(a) R kanala



3.14(b) G kanala



3.14(c) B kanala



3.14(d) RGB irudia

3.14 Irudia: R kanala, G kanala, B kanala eta RGB irudia

3.4 FFmpeg

Datu-baseko bideoak xlsx fitxategiko frame-tartetan banatzeko helburuarekin, FFmpeg [9] software librea erabili da. Baliabide honen bidez, audioak eta bideoak grabatu eta eraldatu daitezke. Nahiz eta sistema eragile gehienetan egon erabilgarri, GNU/Linux-en garatu da.



3.15 Irudia: FFmeg icon

FFmpeg modu anitzetan erabiltzeko aukera eskaintzen du. Aurrera azaldutako prozesuan erabili den egitura orokorra ondorengoa da:

```
ffmpeg -i [sarrera_bideoa] -ss [hasiera_denbora] -t [iraupena] -async 1 -c copy  
[irteera_bideoa]
```

Komando honek lau datu desberdin eskatzen ditu:

[sarrera_bideoa]: zatitu nahi dugun bideoaren path-a.

[hasiera_denbora]: mozketa hasiko den denbora (segundua).

[iraupena]: mozketak izango duen iraupena (segundutan).

[irteera_bideoa]: bideo berriaren izena.

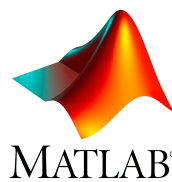
Hurrengo komandoa izanik, "*nire_bideoa.avi*" bideoa zatituko da, 4. segunduan hasita 13 segundu bitartean. Hau da, 4. segundik 17. segundura. Emaidza, "*bideo_berria.avi*" izena izango du.

Komandoa:

```
ffmpeg -i nire_bideoa.avi -ss 4 -t 13 -async 1 -c copy bideo_berria.avi
```

3.5 MATLAB

MATLAB (MATrix LABoratory) datuak aztertu, algoritmoak garatu eta aplikazioak sortzea ahalbidetzen duen softwarea da. Matrizeetan oinarritutako hizkuntza propioa erabiltzen du: M. Programa hau Windows, GNU/Linux, macOS eta Unix sistema eragileetan erabilgarri dago.

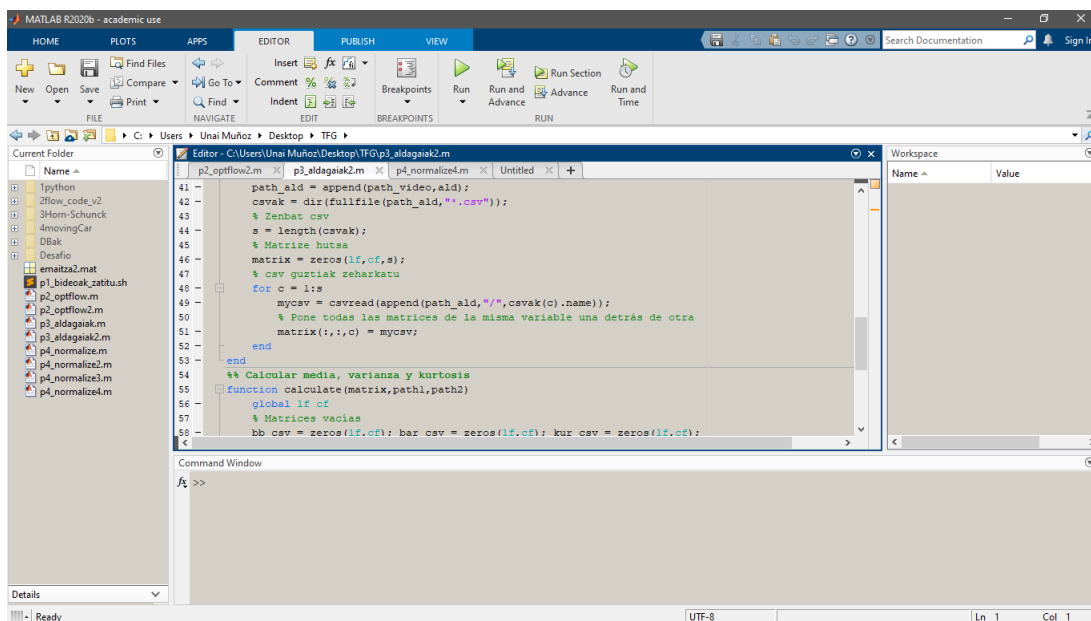


3.16 Irudia: MATLAB icon

MATLAB-ek zeregin garrantzitsua izan du proiektu honetan: bideoen prozesaketaz arduratu da. Horretarako, hiru fitxategi desberdin sortu dira, bakoitza eginkizun finko batekin non garapenaren urrats desberdinak garatzeko erabili diren.

Gainera, alde batetik, bideoez osatutako datu-base batekin modu erosoan lan egin daiteke. Beste aldetik, datuak gordetzeko, eta gero atzitzeko formatu erosoak erabiltzen du. Formatu honek CSV izena du.

Software honek duen interfazea 3.17 Irudian ikusi daiteke.



3.17 Irudia: MATLAB interface

3.6 WEKA

Gainera, WEKA softwarearekin lan egin da. WEKA (Waikato Environment for Knowledge Analysis) ikasketa automatikoan eta datu-meatzaritza oinarritutako ingurunea da. Java programazio-lengoaian inplementatuta dago eta GNU-GPL lizentziapean banatzen den software librea da. Sailkapen, clustering, eta erregresio tresnak eskaintzen ditu, beseen artean.



3.18 Irudia: WEKA icon

Aplikazioa exekutatzean, [3.19a](#) Irudian agertzen den interfazea zabalduko da. Proiektu hau aurrera eramateko, Explorer atala erabili da, [3.19b](#) Irudian agertzen den interfazea erabiliz. Bertan sei atal nagusi ageri dira: Preprocess, Classify, Cluster, Associate, Select attributes eta Visualize. Lan honetan, lehenengo biak erabili dira: lehenengoa aurreprozesaketa burutzeko imageFilters iragazkiak aplikatuz; eta ondoren, bigarrena, sailkatzaileak aplikatu emaitzak lortzeko.

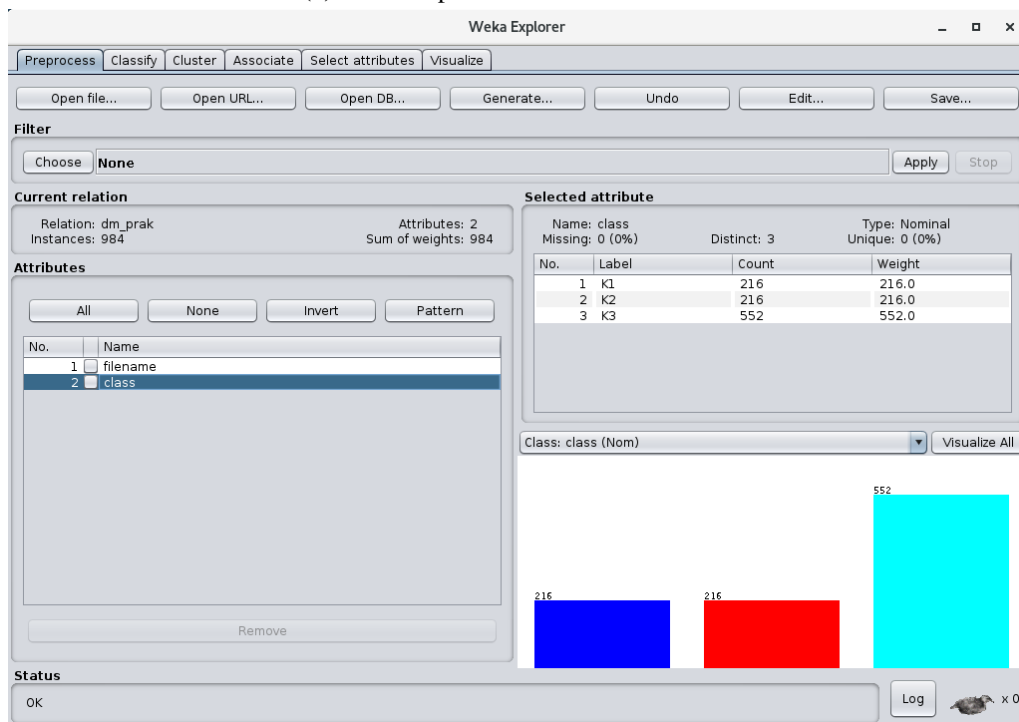
Weka software librea denez, webgunetik deskargatu daiteke. Hori bai, instalatzeko eta exekutzaeko eskakizun batzuk betetzea beharrezkoa da. Eskakizun hauen artean JAVA-ren bertsio zehatza dago.

3.6.1 ARFF formatua

Software honek ARFF (Attribute-Relation File Format) izeneko formatua onartzen du, beste batzuen artean. Bi zati nagusi bereizten dira ARFF fitxategi batean: goiburua eta datuak. ASCII kodeaketa erabiltzen duenez, edozein testu editoreen bitartez sortu, zabaldu eta moldatu daiteke.



3.19(a) WEKA aplikazioaren hasierako interfazea



3.19(b) WEKA aplikazioaren Explorer interfazea

3.19 Irudia: WEKA aplikazioaren interfaze grafikoak

- **Goiburua**

Atal hau fitxategiaren hasieran dago eta bertan erlazioaren izena, atributuen zerrenda eta haien mota ageri da. Erlazioaren izena bakarria izan behar du, baina nahi ditugun atributuak jarri daitezke. Kontuan izan behar da, atributuak informazio eskaintzen dutela, beraz zenbat eta gehiago jarri, gero eta ARFF pisutsua izango dugu. Egitura orokorra:

```
@relation [erlazioaren_izena]
@attribute [atributuaren_izena] [atributu_mota]
```

Atributu motak ondorengoak dira: Numeric (Zenbaki errealak), Integer (Zenbaki osoak), Date (Data formatua), String (Karaktere kateak) eta Enumerate (Atributuak har ditzakeen balio posibleak).

- **Datuak**

Atal hau goiburuaren ondoren jarriko da eta bertan aurretik definitu diren atributuei balioak esleituko zaizkie. Kasu honetan, lerro bakoitza irudi desberdinari egingo dio erreferentzia.

```
@data
[irudiaren_izena] [atributuaren_izena]
```

ARFF adibidea:

```
@relation GrAL
@attribute filename string
@attribute class sitting, standing, walking
@data
sitting_1.png, sitting
sitting_2.png, sitting
...
standing_1.png, standing
standing_2.png, standing
...
walking_1.png, walking
walking_2.png, walking
walking_3.png, walking
...
```

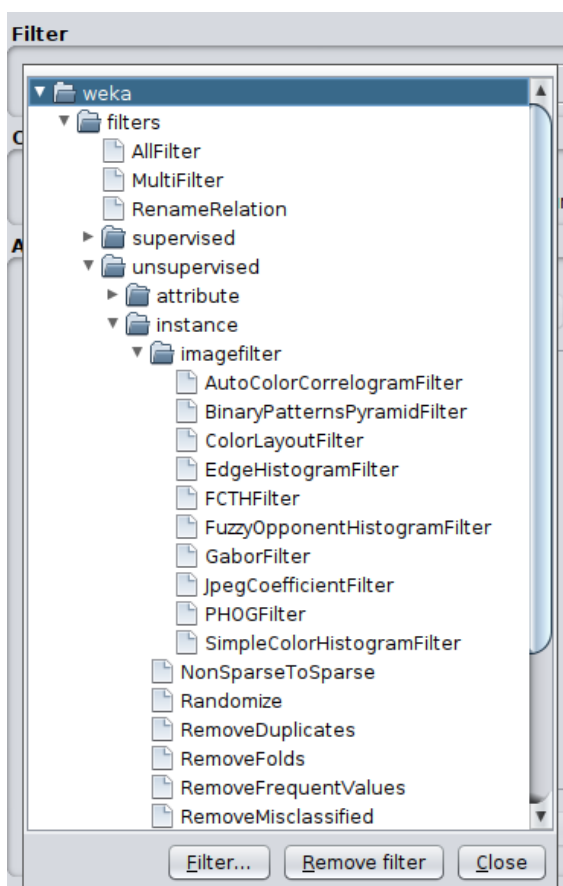
3.6.2 ImageFilters

WEKA softwareak irudi fitxategiak prozesatzeko helburuarekin, hainbat iragazki [10] ditu, zehazki hamar. Iragazki hauek ImageFilters -ak dira eta 3.20 Irudian zerrenda aurkitu dezakegu. Pakete hau eskuratzeko, 3.19a Irudiko "Tools > Package manager" botoietan sakatu ondoren, "ImageFilter" paketea bilatu eta instalatzearekin nahikoa da. Lan honetan lau iragazki erabili dira soilik.

Iragazkiak:

- **AutoColorCorrelogramFilter** (AutoColor): irudi bateko koloreen korrelograma espaziala kalkulatu du.
- **BinaryPatternsPyramid** (Binary): tokiko eredu bitarrak intentsitate eredu bat irudikatzen du (adibidez, ertz bat) puntu baten inguruan. Ondorioz, irudi-eremuetan zehar gertatzen diren eskala handiagoko ereduak kodetzen ditu. Ehundura eta aurpegia ezagutzeko erabilgarriak.
- **ColorLayoutFilter** (ColorLayout): iragazki honek, irudi bat 64 zatitan banatzen du eta zati bakoitzeko batez-besteko kolorea konputatu ostean, ezaugarriak kalkulatu ditu.
- **EdgeHistogramFilter** (Edge): ertzak irudi bateko lerro edo etenaldiak dira. Ertz histograma, ondorioz, ertzek irudian zehar egiten duten jarraibideen laburpena bezala ulertzen da.
- **Fuzzy Color and Texture Histogram** (FCTH): kolore eta ehunduraren informazioa kodetuko du histograma bakarrean. Irudi bakoitzeko 72 bytera mugatua dagoenez, datu-base handietarako egokia da.
- **FuzzyOpponentHistogramFilter** (Fuzzy): aurkako kolore-espazioan oinarrituta dago.
- **GaborFilter** (Gabor): uhin bat erabiltzen du ehundura ateratzeko. Ikuspegi konputazionalerako lanetan oso erabilgarria.
- **JpegCoefficientFilter** (Jpeg): irudi bat JPEG formatura aldatzean, gizakiarentzat hautemaiezina den informazioa baztertzeke joera dago. Gainera, irudiaren irudikapen konprimituaren zati diren koefiziente kuantifikatuen sekuentzia sortzen da. Koefiziente hauek dira iragazki honek konputatzen dituen ezaugarriak.

- **Pyramid Histogram of Oriented Gradients (PHOG)**: irudi bateko intentsitate gradienteen orientazioari buruzko informazioa kodetzen du.
- **SimpleColorHistogramFilter (Simple)**: konputa daitekeen kolore-ezaugarri xumeena. Funtsean, hiru histograma ditu (RGB kolore bakoitzarentzat bat), bakoitzak 32 tarte izanik. Tarte bakoitzak bertan dauden irudien pixel kopurua du.



3.20 Irudia: ImageFilters zerrenda WEKA-n

Aurreprozesaketa egiteko bi modu daude: [3.19b](#) Irudiko interfazea erabiliz edo terminala erabiliz. Lehenengo kasuan, "Open file" eremuan gure ARFF-a aukeratuko dugu eta "Choose" eremuan [3.20](#) Irudiko ImageFilter-etako bat. Amaitzeko, lehenengo atributua (filename) ezabatuko da.

Hona hemen terminal bidezko erabileraren egitura orokorra:

```
java -cp [weka_direktoria] weka.Run [iragazkia] -i [sarrerako_ARFF] -D [fitxategien_direktoria] -o [irteerako_ARFF]
```

[**weka_direktoria**] weka.jar fitxategia non gorde den adierazten du. Adibidez, /home/username/Escritorio/weka.jar

[**iragazkia**] iragazkia aplikatzeko behar den komandoa adierazten du.

[**sarrerako_ARFF**] eta [**irteerako_ARFF**] filter-a aplikatu nahi zaion ARFF-a eta emaitza adierazten dute, hurrenez hurren.

[**fitxategien_direktoria**] datu-baseko irudiak non gordeta dauden adierazten du.

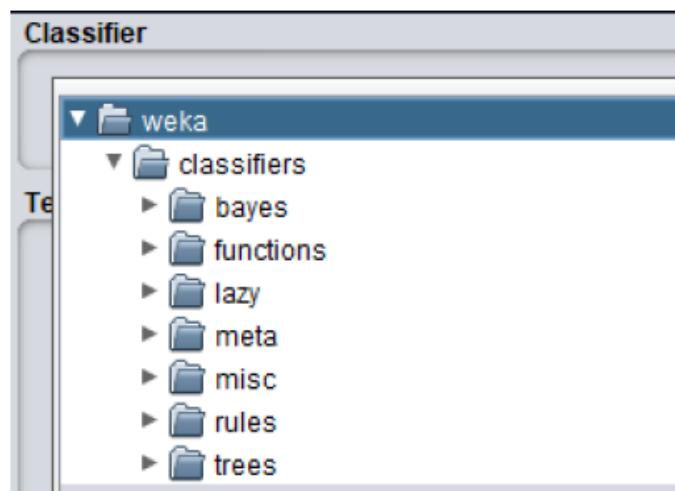
Filter-a aplikatu ondoren, filename izeneko atributua ezabatuko dugu. Kasu honetan ere, bi modura burutu daiteke. Terminalaren bidez komando honekin egingo da:

```
java -cp [weka_direktoria] weka.Run weka.filters.unsupervised.attribute.RemoveType
-T string -i [sarrerako_ARFF] -o [irteerako_ARFF]
```

Lan honetan iragazki hauek aplikatu dira: AutoColorCorrelogramFilter, BinaryPatternsPyramid, ColorLayoutFilter eta Pyramid Histogram of Oriented Gradients.

3.6.3 Sailkatzaileak

Sailkatzaileak [11] irudiak klaseetan sailkatzeko erabiltzen diren algoritmoak dira. Kasu honetan, ekintzetan sailkatzeko. Hainbat motako sailkatzaileak daude eta 3.21 Irudian haien adibideak ikus daitezke (meta izan ezik).

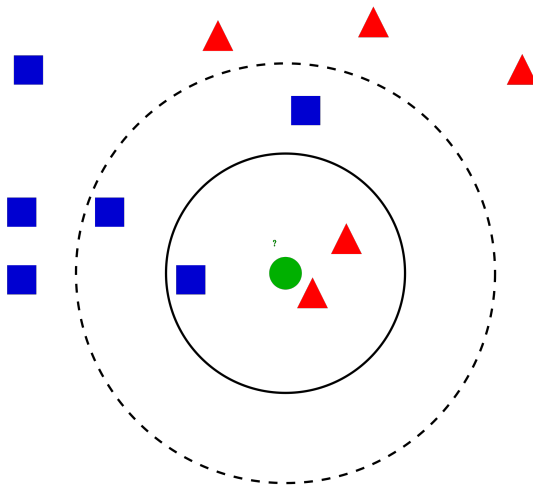


3.21 Irudia: Sailkatzaile motak WEKA-n

- **K-NN**: K auzokide hurbilenetan oinarritzen den sailkatzailea da. Hurbilen dauden K auzokideen balioa aztertzen da eta sarrien agertzen den klasea esleitzen zaio kasu berriari.

3.22 Irudiko kasuan, zirkulu berdea sailkatzeko, K=3 auzokide hurbilenak aztertuz, bi triangelu eta karratu bat hartu dira. Ondorioz, triangelu gehiago daudenez, puntu berdea triangelua dela esango da.

```
java -Xmx2048M weka.classifiers.lazy.IBk -K [K] -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A "weka.core.EuclideanDistance -R first-last" -t [sarrerako_ARFF] -x 10 > [irteerako_ARFF]
```



3.22 Irudia: K-NN, K=3 izanik

- **Naive Bayes (NB)**: Bayesen teoreman oinarritzen da. Sailkatzaile honetan, ezaugarriak elkarren artean independenteak dira klasearen balioa ezaguna denean. Aldagaien independentziaren hipotesiak eragiten duen sinplifikazioari esker, sailkapena egiteko behar diren parametroak estimatzeko datu kopuru txikia behar da.

$$\hat{y} = \operatorname{argmax}_{k \in \{1, \dots, K\}} p(C_k) \prod_{i=1}^n p(x_i | C_k).$$

3.23 Irudia: Naive Bayes eredu probabilistikoa

```
java -Xmx2048M weka.classifiers.bayes.NaiveBayes -t [sarrerako_ARFF] -x 10 > [irteerako_ARFF]
```

- **J48**: C4.5 erabaki zuhaitza sortzen du, inausia ala inausigabea. Horretarako, lehenengo irabazi-ratioak kalkulatu dira eta hauen arabera erroak hautatu. Zuhaitza sortzeko pausuak ondorengoak dira:
 - Oinarrizko kasuak baieztatu.
 - Atributu bakoitzerako irabazi-ratio normalizatua aurkitu.
 - Irabazi-ratio altuena a_best atributua izan dadin.
 - a_best banatzen duen erabaki-nodoa sortu.
 - a_best banaketaren bidez lortutako azpizerrendetan errepikatu, eta nodo horiek seme bezala gehitu.

```
java -Xmx2048M weka.classifiers.trees.J48 -C 0.25 -M 2 -t [sarrerako_ARFF] -x 10 > [irteerako_ARFF]
```

- **RandomForest** (RF): sailkatzaile honek korrelaziorik gabeko zuhaitzen bilduma eraikitzen du, ausaz aukeratutako balioen arabera. Bagging-aren moldaketa bat da, zuhaitzen bilduma hori eraikitzen duena, eta ondoren, batezbestekoak kalkulatu dituen.

```
java -Xmx2048M weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -t [sarrerako_ARFF] -x 10 > [irteerako_ARFF]
```

- **SVM** (SMO): algoritmo hau iteratiboa da, problemak azpi problema txikiagoetan banatu eta ondoren analitikoki konpontzen dituen. Funtzio koadratikoaren minimoa aurkituko da problema analitikoki ebazteko.

```
java -Xmx2048M weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007-calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4-t [sarrerako_ARFF] -x 10 > [irteerako_ARFF]
```

3.6.4 Multi-sailkatzaileak

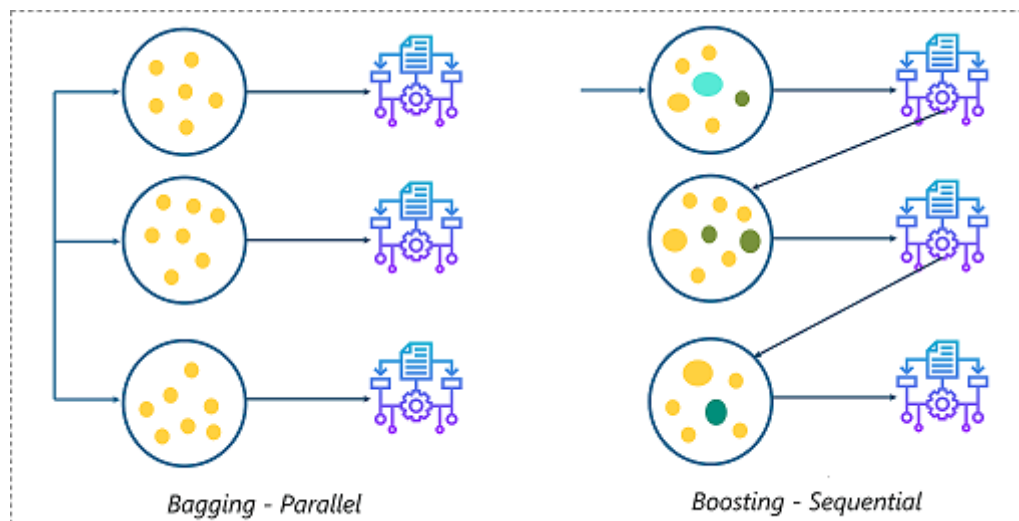
Emaitza hobek ateratzeko helburuarekin, zenbait sailkatzaile konbinatzen dira, sailkatzaile berriak eratuz. Hipotesi nagusia hurrengoa da: eredu ahulak batzea, eredu indartsuagoa lortu dadin. Sailkatzaile hauek 3.21 Irudiko meta karpetan aurkitzen dira.

- **Bagging:** teknika hau N ereduko multzo bat eraikitzean oinarritzen da. Multzo bakoitza eraikitzeke, entrenamenduko datu guztiak errepikatuko dituen laginketa bat eginez burutuko da.

```
java -Xmx2048M weka.classifiers.meta.Bagging -P 100 -S 1 -num-slots 1 -I 10 -W [bigarren_sailkatzailea] -t [sarrerako_ARFF] -x 10 > [irteerako_ARFF]
```

- **Boosting:** teknika honek Bagging teknika baino estrategia burutsuagoa erabiltzen du. Iterazio bakoitzak aurretik egindako akatsak zuzentzen saiatzen baita gaizki sailkatu diren datuei pisu handiagoa emanez.

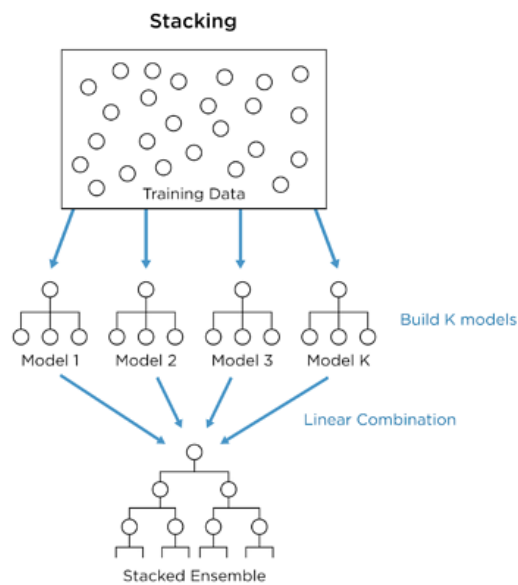
```
java -Xmx2048M weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -num-slots 1 -I 10 -W [bigarren_sailkatzailea] -t [sarrerako_ARFF] -x 10 > [irteerako_ARFF]
```



3.24 Irudia: Bagging eta boosting

- **Stacking**: N oinarrizko sailkatzaile eraikitzen ditu, algoritmo bera, baina entrenamendu-multzo desberdinak erabiliz. Iragarpenak bere oinarrizko sailkatzaileen iragarpenetatik egiten ditu, boto gehien dituen klasea emaitza bezala hartzen delarik.

```
java -Xmx2048M weka.classifiers.meta.Stacking -X 10 -M [bigarren_mailako_sailkatzailea] [lehenengo_mailako_sailkatzaileak] -t [sarrerako_ARFF] -x 10 > [irteerako_ARFF]
```



3.25 Irudia: Stacking

4. KAPITULUA

Garapena

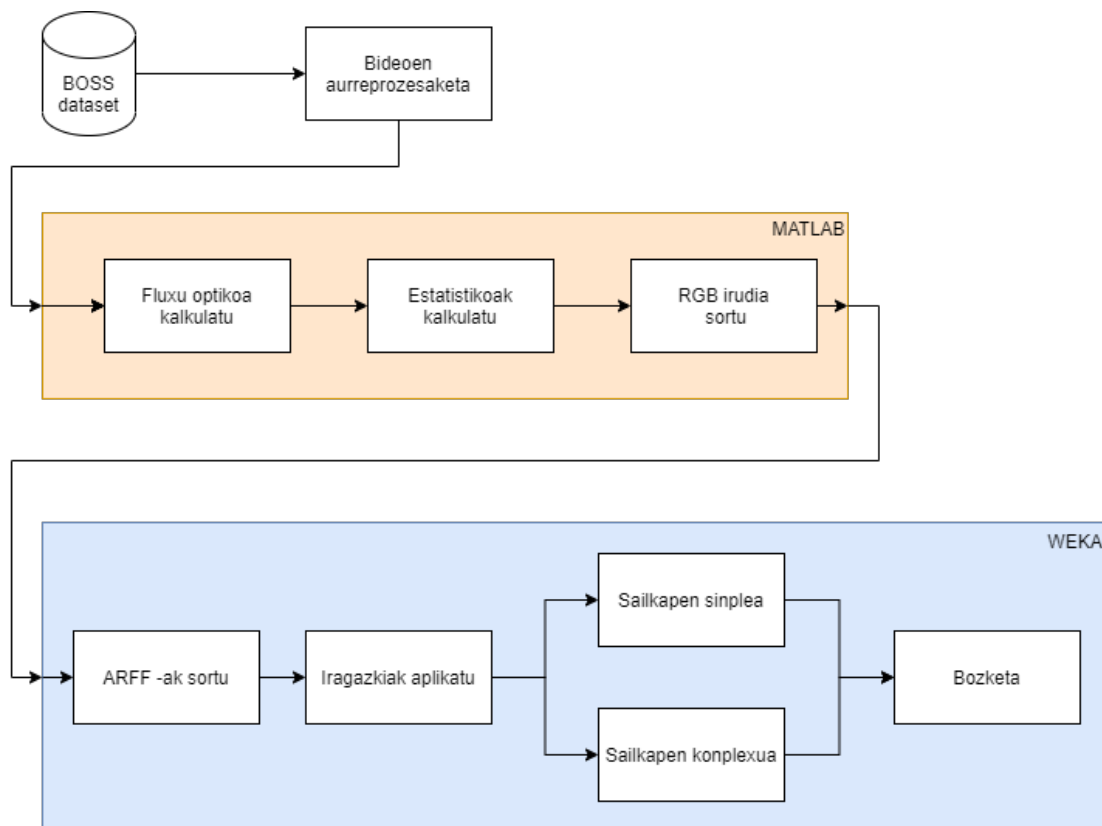
Atal honetan, proiektuaren garapena azalduko da, hau da, hasieratik bukaerara burutu diren urrats guztiak. Horretarako, hiru atal nagusi bereizi dira, bakoitza ingurune desberdinean garatua.

- Bideoen aurreprozesaketa
 - Bideoak zatitu
- Bideoen analisisa
 - Fluxu optikoa kalkulatu
 - Estatistikoak kalkulatu
 - RGB irudia sortu
- Bideoen aurreikuspena
 - ARFF-a sortu
 - Iragazkiak aplikatu
 - Sailkapen sinplea
 - Sailkapen konplexua
 - Bozketa

Ondoren, 4.1 Irudian, prozesu honen eskema orokorra ageri da, atal bakoitza maila desberdinean eta kolore desberdinez adierazita. Lehenengo maila, pausu bakar batez osatuta dago, FFmeg softwarea erabiliz burutu dena. Bigarren maila (kolore laranja margotuta) MATLAB-en garatu da eta, hirugarrena (urdinez margotuta), aldiz, WEKA-n.

Orokorrean prozesu lineala jarraitu da, hau da, edozein pausu burutzeko aurrekoa amaituta egotea beharrezkoa izan da. Hala eta guztiz ere, bi sailkapen mota egin direnez, kasu horretan bata ez du bestearen beharra izan. Ondorioz, zati hori paraleloan burutzea posible izan da.

Prozesua Bozketa egiterakoan bukatuko da. Hau horrela bada ere, aurreko pausuan ere emaitza batzuk lortzen dira. Ondorioz, prozesuaren hiru zatitan emaitzak aztertzeko eta ondorioak ateratzeko momentua izango dugu: sailkapen sinplea, konplexua eta hauen bien bozketa egiterakoan.



4.1 Irudia: Garapena

4.1 Bideoen aurreprozesaketa

Lehenengo atalak urrats bakarra du eta helburua eskuratu dugun BOSS datu-basea gure beharretara moldatzean datza: gure datu-baseko bideoak zatitzea edukiaren arabera. Proiektuaren zati hau Linux-en garatu da, eta "*p1_bideoak_zatitu.sh*" izeneko script-ak horren ardura izango du. Script-a fitxategi bat da, Linux-eko aginduez osatuta dagoena, eta sistemak orden horretan exekutako dituen.

4.1.1 Bideoak zatitu

Arestian aipatu bezala, *xlsx* fitxategiko datuen arabera, bideoak zatituko dira datu-basea handituz. 3.4 atalean azaldu den bezala, FFmpeg softwareak denbora segundutan eskatzen duenez eta fitxategi honetan frame-tarteak ageri direnez, moldaketa egin beharko da modu zuzenean exekutatzeke.

Modu honetan, script-ari lau argumentu pasako zaizkio: zatitu nahi den bideoaren izena, sortuko den bideo berriaren izena, mozketaren hasierako eta bukaerako framea. Script-ak datu horiek moldatuko ditu, FFmpeg komandoari egokituz. Bideo guztiak irudiez osatuta daude, fotograma edo frame izenekoak. Giza begia soilik segunduko 10-12 frame detektatzeko gai da, eta hemendik gorako fotograma sekuentziak mugimendua bezala hautematen dira. Normalean, bideoak segunduko 25 framez osatuta daude. Honako hau da komandoaren egitura orokorra:

```
sh p1_bideoak_zatitu.sh [sarrerako_bideoa] [irteerako_bideoa] [hasierako_frame]  
[bukaerako_frame]
```

Adibide gisa "*Cell_phone_Spanish.Cam01.avi*" bideoa hartuz, 3.2 Irudiko datuaen arabera zatituko dugu. Jarraian dauden komandoekin zatitu da bideo hau:

Lehenik eta behin, lehenengo pertsona, [414,481] eta [1282,1328] tartetan ibiltzen ari da. Honen arabera, badakigu 16,56 segunduan hasi dela ibiltzen eta 19,24 segunduan bukatu (2,68 segundu bitartean). Gainera, 51,28 segunduan berriro ibiltzen hasi da eta 53,12 segunduan bukatu (1,84 segundu bitartean).

Datu hauek kontuan hartuz lehenengo bideoaren lehenengo zatiketak ondorengo komandoekin exekutatu dira:

```
sh p1_bideoak_zatitu.sh Cell_phone_Spanish.Cam01.avi wal_cps_c01_v01.avi 414  
481
```

```
sh p1_bideoak_zatitu.sh Cell_phone_Spanish.Cam01.avi wal_cps_c01_v02.avi 1282  
1328
```

Sartutako datuak moldatu ostean, ondorengo FFmeg komandoak exekutatu dira script bakoitzean.

```
ffmpeg -i p1_bideoak_zatitu.sh -ss 16,56 -t 2,68 -async 1 -c copy wal_cps_c01_v01.avi
```

```
ffmpeg -i p1_bideoak_zatitu.sh -ss 51,28 -t 1,84 -async 1 -c copy wal_cps_c01_v02.avi
```

Gainera, bigarren pertsona, [725,835], [990,1057] eta [1091,1129] tartetan ibiltzen ari dela badakigu, eta komando hauekin zatituko dira hiru tarte hauek:

```
sh p1_bideoak_zatitu.sh Cell_phone_Spanish.Cam01.avi wal_cps_c01_v03.avi 725  
835
```

```
sh p1_bideoak_zatitu.sh Cell_phone_Spanish.Cam01.avi wal_cps_c01_v04.avi 990  
1057
```

```
sh p1_bideoak_zatitu.sh Cell_phone_Spanish.Cam01.avi wal_cps_c01_v05.avi 1091  
1128
```

Zatiketarekin jarraituz, pertsona biak [482,1110] eta [836,989] tartetan eserita daude, hurrenez hurren. Frame-tarte hauek lehenengo bideoako zatirik luzeenak sortu dute:

```
sh p1_bideoak_zatitu.sh Cell_phone_Spanish.Cam01.avi sit_cps_c01_v01.avi 482  
1110
```

```
sh p1_bideoak_zatitu.sh Cell_phone_Spanish.Cam01.avi sit_cps_c01_v02.avi 836  
989
```

Amaitzeko, [1111,1282] eta [1058,1090] tartetan lehenengo eta bigarren pertsona altxatuta daude, hurrenez hurren.

sh p1_bideoak_zatitu.sh Cell_phone_Spanish.Cam01.avi sta_cps_c01_v01.avi 1111
1282

sh p1_bideoak_zatitu.sh Cell_phone_Spanish.Cam01.avi sta_cps_c01_v02.avi 1058
1090

Sortu diren bideo berriak modu ordenatuan sailkatu daitezten, patroia bat jarraituko da nomeklatura zehazteko. Nomeklatura hau, klasearen eta kameraren arabera zehaztuko da. Ondorioz, izena irakurriz, bideoaren edukia jakiteko aukera izango dugu:

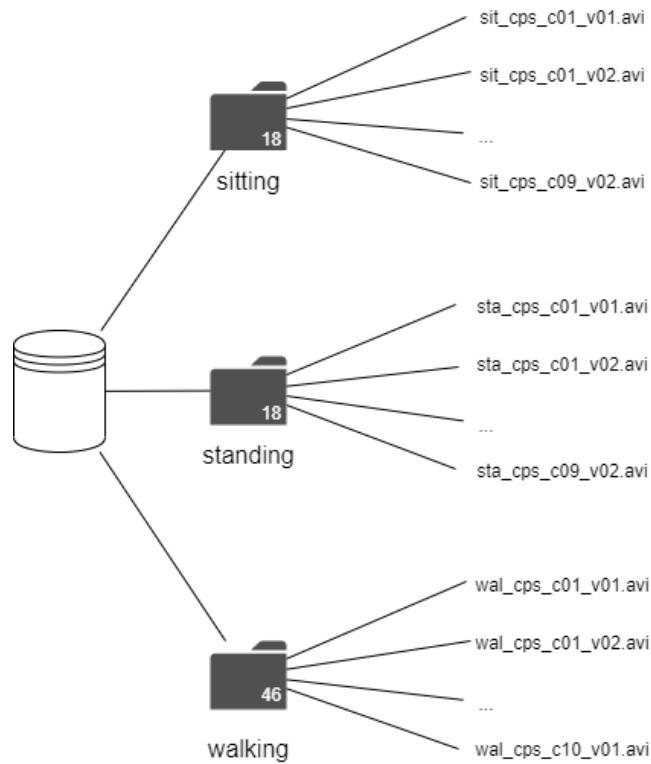
- **Klasea:** lehenengo hiru karaktereek (KKK bidez adierazita) klasea adierazten dute. 'sit' sekuentziak *sitting/eseri* klaseari egingo dio erreferentzia, 'sta' *standing/altxa* klaseari eta 'wal' *walking/ibili* klaseari.
- **Kamera:** XX karaktereek kamera zenbakia adierazten dute, lehenengoa '01' kamera eta azkenengoa '10' kamera izanik.
- **Bideoa:** klase eta kamera berdineko hainbat bideo sortu daitezkeela kontuan hartuz, YY karaktereek bideo zenbakia adierazten dute.

Laburbilduz, hurrengoa da bideo berrien egitura orokorra:

KKK_cps_cXX_vYY.avi

Modu honetan, alde batetik "*sit_cps_c01_v02.avi*" bideoa izanik, *sitting* klasearen, lehenengo kamerako bigarren bideoa izango da. Beste aldetik, "*wal_cps_c03_v04.avi*" bideoa *walking* klasearen, hirugarren kamerako laugarren bideoa da.

Bideo guztiak ordenatuta egoteko helburuarekin, klase bakoitzeko karpeta bat sortu da eta bertan bakoitzari dagokion bideoak. 4.2 Irudian ageri den egitura sortu da zatiketa guztiak egin eta gero. *walking* klaseko 46 bideo sortu dira, eta *sitting* eta *standing* klaseko 18 bideo bakoitzean, guztira 82 bideo.



4.2 Irudia: New dataset



4.3 Irudia: SITTING klaseko bideoen aurreikuspena

4.2 Bideoen analisisia

Prozesuarekin jarraituz, garapenaren atalik garrantzitsuenean, datu-base berriko bideoekin lan egingo da hiru pauso desberdinetan. Atal hau garapenaren bigarren mailari dagokio eta MATLAB-en garatu da. 4.1 Irudian ageri den bezala, atal hau hiru pausotan banatzen da, bakoitza fitxategi desberdinean garatu dena.

Atalaren hasieran, 4.2 Irudiko datu-basea erabiliko dugu amaieran RGB irudiak izateko helburuarekin. Bitartean, fluxu optikoko teknikak erabili dira.

Erabilitako fitxategiak:

- *"p2_optflow2.m"*
- *"p3_aldagaiak2.m"*
- *"p4_normalize4.m"*

4.2.1 Fluxu optikoa kalkulatu

Atal honetako lehenengo pausua, datu-base berriari fluxu optikoa aplikatzean datza. *"p2_optflow2.m"* fitxategia exekutatzuz lortu da ataza hau.

Fluxu optikoa irudi bateko pixelen higidura bezala definitzen da, irudian proiektatzen den mugimendua ikustaraziz. Teknika hau erabiltzen dituzten metodo gehienek, pixel baten kolorea eta intentsitatea desberdina dela bermatzen dute frame batetik bestera.

Horn-Schunck fluxu optikoaren algoritmoak [12] leuntze-konpresioak sartzen ditu irudian dagoen mugimendua estimatzeko. Ereku argi batetik, ereku ilun batera bitarteko aldaketa pixkana izan behar du, bitarteko intentsitateetatik igaroz. Modu berean, pixel bateko intentsitatea, hurrengo irudiko pixel berdinarekin intentsitate aldaketa leuna izan behar du.

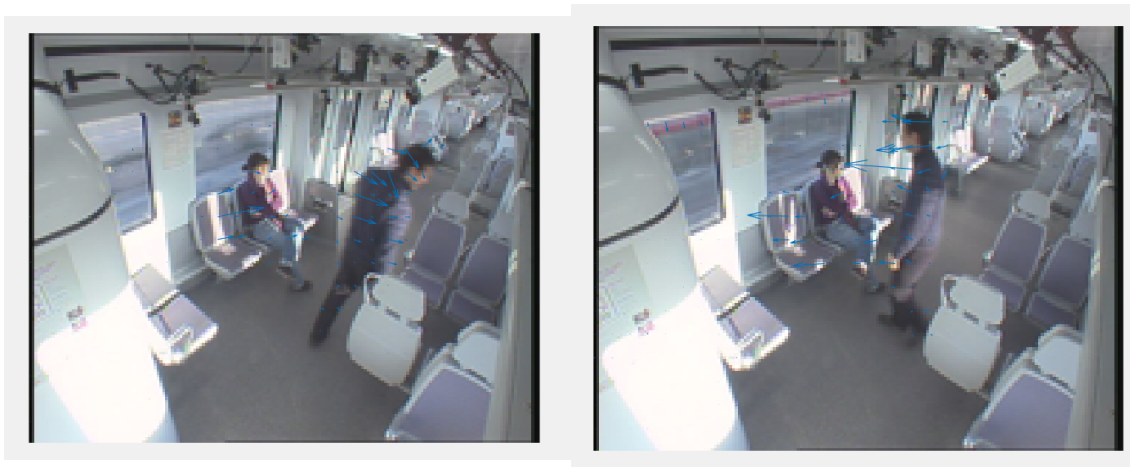
Ondorengo 4.4 Irudian Horn–Schunck fluxu optikoaren ekuazioa ageri da.

$$\int \int \left[\left(I_x u + I_y v + I_t \right)^2 + \alpha^2 \left(\|\nabla u\|^2 + \|\nabla v\|^2 \right) \right] dx dy$$

4.4 Irudia: Horn–Schunck fluxu optikoaren ekuazioa

non $I_x u$, $I_y v$ eta I_t x, y eta t denbora dimentsioen arabeko intentsitate balioen deribatuak diren, hurrenez hurren.

Ausaz bi frame hartuz, eta haien fluxu optikoa kalkulatzean aurreko bi emaitzak lortu dira. 4.5a Irudian ageri den gizona, eskumarako norantza duela ikusi dezakegu geziak eskumara begira daudelako. Esertzeko asmoarekin, burua behera mugitzen ari da, eta hori gezien norabideari esker ondorioztu da. 4.5b Irudian, ordea, ezkerrera ibiltzen ari da, emakumearen norantzan. Gainera, nabaria da abiadura handiagoa dela, gezien magnitudea dela eta. Bi kasuetan, nahiz eta emakumea eserik egon, aldaketak detektatu dira haren inguruan: eskuak eta burua mugitzen ditu modu diskretuan.



4.5(a) Eskumarako fluxu optikoa

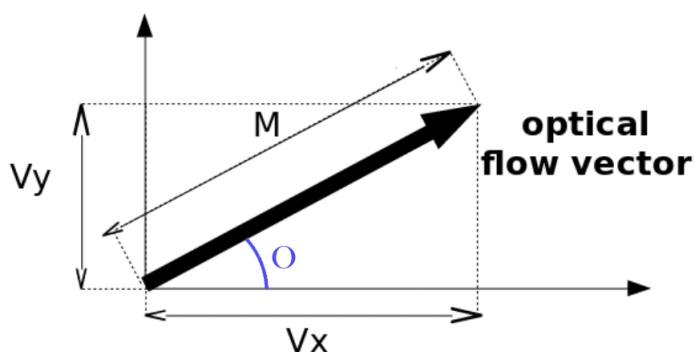
4.5(b) Ezkerrerako fluxu optikoa

4.5 Irudia: Fluxu optikoaren adibidea

Teknika hau bideo guztien frame guztiei aplikatu zaie, aurreko framearekin alderatuz. Kalkulu bakoitzean, fluxu optikoak lau aldagai desberdinetan gordetzen ditu datuak, bakoitza matrize desberdinean. Modu honetan, lau matrizeetako koordinatu berdineko balioek pixel berdina deskribatzen dute.

Fluxu optikoaren osagaiak 4.6 Irudian irudikatuta daude:

- **Abiadura (V_x)** → Fluxuaren norabidea abzisa (x) ardatzean.
- **Abiadura (V_y)** → Fluxuaren norabidea ordenatu (y) ardatzean.
- **Magnitude (M)** → Fluxuren luzera.
- **Orientation (O)** → Fluxuaren norabidea.



4.6 Irudia: Optical flow

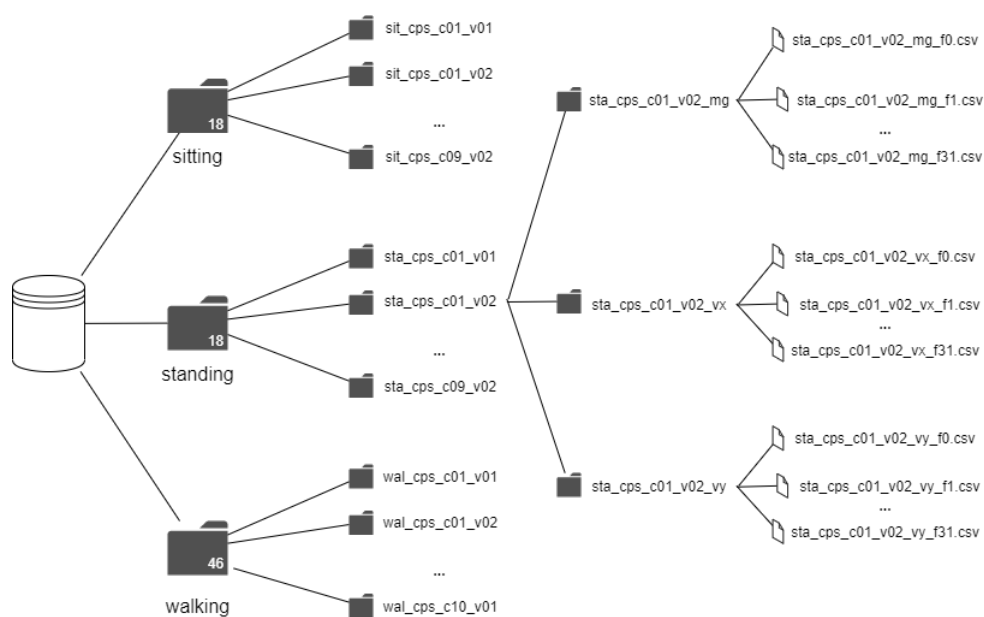
Hau guzti hau kontuan hartuta, hurrengo ondorioztatu dugu: mugimenduaren norabidea, V_x eta V_y matrizeetan gordetzen da, aldaketa zein noranzkoan gertatu den adieraziz. Mugimenduaren aldaketaren indarra, magnitude matrizeetan gordetzen da, zenbateko aldaketa gertatu den adieraziz.

Orientation (O) eta bi ardatzetako norabideek (V_x eta V_y) erlazio zuzena dute: bi norabideek sortzen duten angelua, orientation denez, trigonometriaren arauak aplikatuz ondorengo betetzen da: orientationaren tangentea, abiaduren arteko zatiketa da. Beste modu batera esanda, norabideen arteko zatiketaren arkotangentea, orientation da.

$$\tan(O) = \frac{V_y}{V_x} \quad O = \arctan\left(\frac{V_y}{V_x}\right)$$

Hori dela eta, orientation aldagaia alde batera utziko dugu; ez baitu informazio berririk eskaintzen. Ondorioz, bideo bakoitzeko frame guztien fluxu optikoa kalkukatu ostean, V_x , V_y eta Magnitude aldagaien informazioa eskuratu da CSV fitxategietan gordeta. Datu guzti hauek gordetzeko datu-base berria osatuko da.

4.2 datu-basean bezala, klase bakoitzeko karpeta bat sortu da, haien barnean bideo bakoitzeko beste bat. Oraingoan, aldagaia bakoitzeko beste karpeta bat sortu da, hau da, Vx, Vy eta Magnitude-rako karpeta bana. 4.7 Irudian datu-base berriaren egitura ikusi daiteke. Hain zuzen ere, *standing* klasearen "*sta_cps_c01_v02.avi*" bideoaren egitura azaltzen da. Oraingoan, irudiak izan beharrean csv fitxategiak ditugu, informazioa matrize eran gordeta.



4.7 Irudia: Optical flow CSV dataset

Kasu honetan ere, CSV fitxategiak modu ordenatua gorde daitezkeen, patroia bat zehaztu-ko dugu. Fitxategiak bideoa sailkatu direnez, izenaren lehenengo zatiak haien patroia jarraituko dute, eta amaitzeko patroia berria adieraziko da:

- **Klasea:** lehenengo hiru karaktereek (KKK bidez adierazita) klasea adierazten dute. 'sit' sekuentziak *sitting/eseri* klaseari egingo dio erreferentzia, 'sta' *standing/altxa* klaseari eta 'wal' *walking/ibili* klaseari.
- **Kamera:** XX karaktereek kamera zenbakia adierazten dute, lehenengoa '01' kamera eta azkenengoa '10' kamera izanik.
- **Bideoa:** klase eta kamera berdineko hainbat bideo sortu daitezkeela kontuan hartuz, YY karaktereek bideo zenbakia adierazten dute.

- **Aldagaia:** AA karaktereek aldagaia adierazten dute. Hiru aukera posible daude: 'vx' karaktere sekuentziak Vx aldagaiari egingo dio erreferentzia, 'vy' Vy aldagaiari eta 'mg' *Magnitude* aldagaiari.
- **Frame:** ZZ karaktereek frame zenbakia adierazten dute, '00' -tik hasiz. Bideoaren luzeraren arabera, azkenengo datua aldatuko da.

KKK_cps_cXX_vYY_AA_fZZ.csv

Modu honetan, "sit_cps_c01_v02_vx_f23.csv" fitxategia izanik, *sitting* klasearen, lehenengo kamerako bigarren bideoa aztertu da eta Vx aldagaiaren 23. frame-eko datuak ditu.

4.2.2 Estatistikoak kalkulatu

Atal berdinean jarraituz, bideo bakoitzeko hiru aldagaietan oinarritutako fluxu optikoa eskuratu ondoren, datu horiek bildu behar dira, datu-basea sinplifikatuz. Horretarako, parametro estatistiko batzuk aplikatuko dira. "p3_aldagaiak2.m" fitxategia exekutatzeko lortu da ataza hau.

Parametro estatistikoek datu multzo baten balio adierazgarriak kalkulatu dituzte. Horrela, ordezkari moduko emaitza lortzen da, eta hori interpretatuz datuen laburpen modukoa lortzen da. Hona hemen erabilitakoak:

- **Batezbestekoa:** Balio kuantitatiboez osatutako datu multzoa kontuan izanik, erdiguneko balio bezala ezaguna da. Balio guztiak batuz eta kopuruarekin zatituz lortzen da.

$$\bar{x} = \frac{\sum_i x_i}{n}$$

- **Bariantza:** Datu multzo batek duen sakabanatzearen neurri absolutua da. Hain zuzen, bariantzaren erro karratu positiboa desbideratze estandarra da, eta azken honek datu bakoitza batezbesteko aritmetiko sinpletik zenbat desbideratzen den adierazten du.

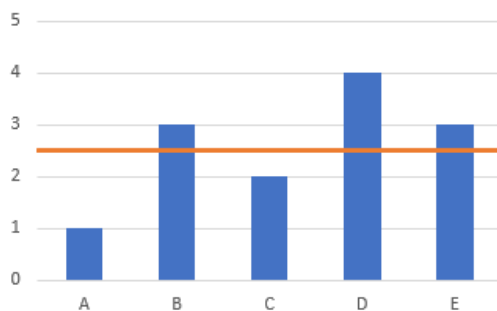
$$s_x^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

- **Kurtosia:** Banaketa baten zorrotasun maila adierazten du. Kurtosi handia duen banaketa zorrotzagoa izango da batezbestekoaren inguruan. Kurtosi txikiagoko banakuntzek, berriz, zentro zapala eta mutur labur eta arinagoak dituzte.

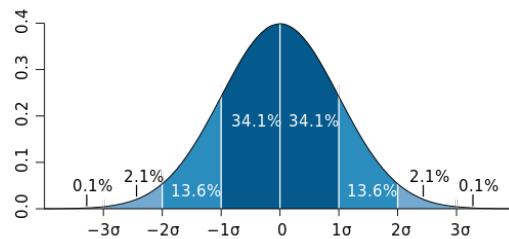
Hona hemen bi zenbaki segida eta bakoitzaren parametro estatistikoaren kalkulak:

	[1,2,13,8,5]	[4,6,8,5,4,5]
Batezbestekoa	5.8	5.333333
Bariantza	23.7	2.266667
Kurtosia	1.535801	2.223183

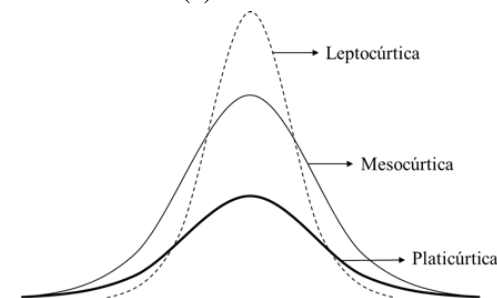
4.1 Taula: Parametro estatistikoaren adibideak



4.8(a) Batezbestekoa



4.8(b) Bariantza



4.8(c) Kurtosis

4.8 Irudia: Parametro estatistikoaren grafikoak

Bideo bakoitzetik hiru aldagai lortu direnez, hiru datu multzo lortu ditugu. Datu multzo bakoitzean hiru parametro estatistiko horiek kalkulatu dira: batezbestekoa, bariantza eta kurtosia. Datu multzoa matrizeez osatuta dagoenez, emaitza matrize bat izango da. Koordinatuz koordinatu balioak hartuko dira, eta kalkulak egin eta gero, matrize berriaren koordinatu berdinean gordeko dira.

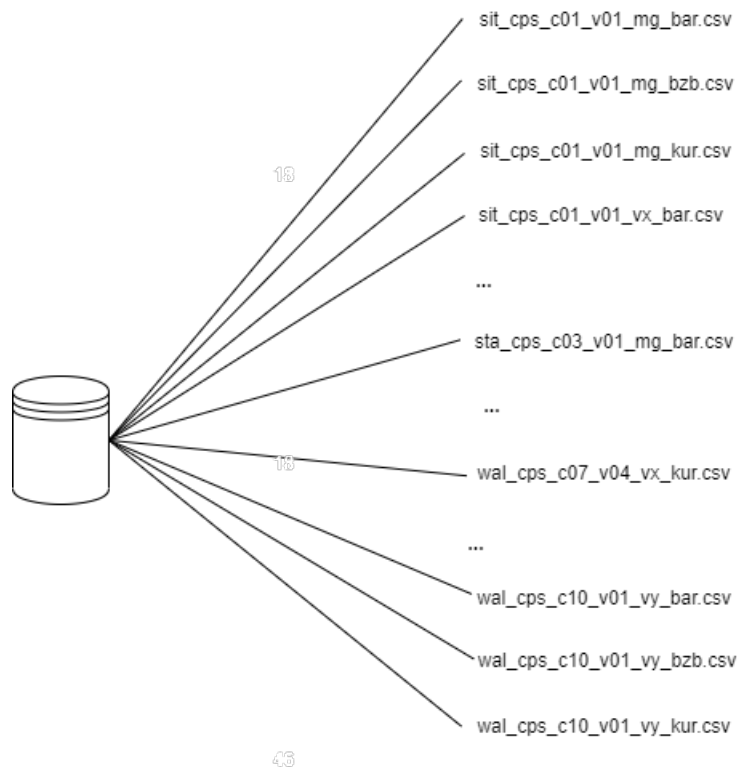
Prozesu honen amaieran, frame guztietako informazioa fitxategi bakarrean gordeta izango dugu, hiru bider. Hau da, bideo bakoitzetik hiru aldagaiko datuak ditugu, eta bakoitzean hiru kalkulu egin dira. Ondorioz, bideo bakoitzeko guztira 9 CSV izango ditugu.

Ondorengo 4.2 Taulan, "sit_cps_c01_v01.avi" bideotik lortu diren 9 CSV-ak ditugu. Zutabeka hiru aldagaiak ditugu eta lerro bakoitzean parametro estatistikoak. Modu honetan, adibidez, sit_cps_c01_v01_vx_bar.csv -ak Vx aldagaiaren bariantzako kalkuluak gordetzen ditu.

	Vx	Vy	Magnitude
BB	sit_cps_c01_v01_vx_bzb	sit_cps_c01_v01_vy_bzb	sit_cps_c01_v01_mg_bzb
Bar.	sit_cps_c01_v01_vx_bar	sit_cps_c01_v01_vy_bar	sit_cps_c01_v01_mg_bar
Kurt.	sit_cps_c01_v01_vx_kur	sit_cps_c01_v01_vy_kur	sit_cps_c01_v01_mg_kur

4.2 Taula: "sit_cps_c01_v01.avi" bideoaren 9 CSV-ak

Oraingoan datu-base moldatu da berriro. Fitxategi guztiak direktorio berdinean gorde dira, eta erabilitako nomeklatura dela eta modu ordenatuan sailkatu dira. Hasierako zatiketa egin ondoren 82 bide sortu ziren, eta bakoitzetik 9 kalkulu egin direnez, 738 fitxategi izango ditugu urrats hau eta gero.



4.9 Irudia: RGB CSV dataset

4.2.3 RGB irudia sortu

Bigarren ataleko azkenengo urratsean, RGB irudiak sortuko dira. "*p4_normalize4.m*" fitxategia exekutatuaz lortu da ataza hau.

Helburua aurreko ataleko CSV fitxategietatik RGB irudiak sortzea da. Bideo bakoitzetik, 9 CSV fitxategi lortu direnez, hiru hiru hartuko dira hiru geruzako hiru bideo osatzeko. Azaldu bezala, fitxategi hauek matrize itxura dutenez, balio bakoitza geruzako pixel bati dagokio. Batezbesteko datuak R kanalatzat hartuko dugu, bariantza datuak G kanalatzat eta kurtosi datuak B kanalatzat. Modu honetan, hiru geruza horiek elkartuz, RGB irudia sortuko da.

R kanala	sit_cps_c01_v01_vx_bzb.csv
G kanala	sit_cps_c01_v01_vx_bar.csv
B kanala	sit_cps_c01_v01_vx_kur.csv

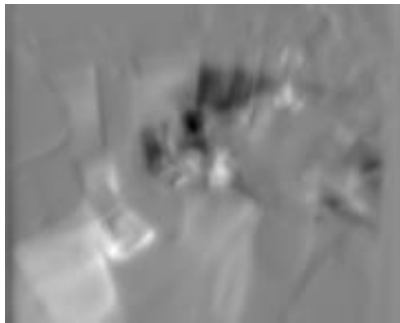
4.3 Taula: "*sit_cps_c01_v01.avi*" bideoaren Vx aldagaiaren geruzak

R kanala	sit_cps_c01_v01_vy_bzb.csv
G kanala	sit_cps_c01_v01_vy_bar.csv
B kanala	sit_cps_c01_v01_vy_kur.csv

4.4 Taula: "*sit_cps_c01_v01.avi*" bideoaren Vy aldagaiaren geruzak

R kanala	sit_cps_c01_v01_mg_bzb.csv
G kanala	sit_cps_c01_v01_mg_bar.csv
B kanala	sit_cps_c01_v01_mg_kur.csv

4.5 Taula: "*sit_cps_c01_v01.avi*" bideoaren mg aldagaiaren geruzak



4.10(a) R kanala



4.10(b) G kanala



4.10(c) B kanala



4.10(d) RGB irudia

4.10 Irudia: R, G, B kanalak eta RGB irudia

R kanala	sit_cps_c01_v01_vx_r.jpg
G kanala	sit_cps_c01_v01_vx_g.jpg
B kanala	sit_cps_c01_v01_vx_b.jpg

4.6 Taula: "sit_cps_c01_v01.avi" bideoaren Vx aldagaiaren irudiak

R kanala	sit_cps_c01_v01_vy_r.jpg
G kanala	sit_cps_c01_v01_vy_g.jpg
B kanala	sit_cps_c01_v01_vy_b.jpg

4.7 Taula: "sit_cps_c01_v01.avi" bideoaren Vy aldagaiaren irudiak

R kanala	sit_cps_c01_v01_mg_r.jpg
G kanala	sit_cps_c01_v01_mg_g.jpg
B kanala	sit_cps_c01_v01_mg_b.jpg

4.8 Taula: "sit_cps_c01_v01.avi" bideoaren mg aldagaiaren irudiak

Aipatzekoa da datu horiek $[0,1]$ tarteko balioak direla eta RGB formatuak $[0,255]$ balioko datuak hartzen dituen bezala, normalizatzea beharrezkoa da, koloreak galdu ez daitezen. RGB irudia ez ezik, geruza bakoitzak sortzen duen irudia ere gordeko dugu.

Goiko irudian gris eskalan dauden [4.10a](#) Irudia, [4.10b](#) Irudia eta [4.10c](#) Irudia elkartuz, [4.10d](#) Irudia eratuko da, koloreak sortuz.

4.3 Bideoen aurreikuspena

Azkenengo atala lau azpiataletan banatu da. Maila honen helburu nagusia orain arte lortutako irudiak sailkatzea da, bertatik emaitzak eta ondoriak ateratzeko. Bi sailkapen mota aplikatu dira, eta ondoren bozketan oinarrituako prozesua aplikatu da. Lehenengo azpiatala beste hiruretan lan egiteko prestakuntza da. Hori dela eta, lehenengo izan ezik, beste guztiak WEKA-n garatu dira.

Aipatatuko bi sailkapenak modu independentean aplikatzea posiblea da, baita bakarrarekin lan egitea. Hala eta guztiz ere, biak aztertzea oso interesgarria izan da haien arteko desberdintasuna alderatzean. Gainera, biak aplikatuz emaitza gehiago lortuko direnez, bozketan ere, ondorio gehiago ateratzea posible izango da.

4.3.1 ARFF-a sortu

WEKA-rekin lanean hasi baino lehen, ARFF fitxategia prestatu beharko dugu. Hasteko, aurreko mailan lortutako datu-basea, irudiez osatuta dagoena, oinarri moduan erabiliko dugu. Kasu honetan, irudi guztiak direktorio berdinean aurkitzen dira.

Kasu honetan, helburua, fitxategi bat sortzea da, irudi guztien informazioa gordeko duena WEKA irakurri dezan. Horretarako, script bat erabili da lan hau automatikoki egiteko. Script-ak *"DB2arff.sh"* izena du eta datu-basearen path-a argumentu bezala sartu behar zaio.

```
sh DB2arff.sh rgb_irudiak
```

Programa honek, ARFF formatua duen fitxategi bat sortuko du "irteera" izeneko direktorioan eta irudi guztiak karpeta bakarrean gordeko ditu, dagokion izenarekin. Gainera, fitxategi honek datu-basearen informazio guztia gordeko du.

Hasteko datu-basearen erlazio edo izenburua "GrAL" moduan izendatuko da. Lehenengo atributua string bat izango da, fitxategiaren izena izango duena. Bigarren atributua, aldiz, klasearen informazioa eskaintzen du. Klaseen izena sinplifikatzeko, hemendik aurrera "sitting" klasea "K1" izango da, "standing" klasea "K2" eta "walking" klasea "K3". Ondoren, @data atala agertuko da.

"K1" klasearen lehenengo bideoa "K1-10.jpg" eta azkena "K1-225.jpg" da. "K2" klasearen lehenengo bideoa "K2-10.jpg" eta azkena "K2-225.jpg" da. "K3" klasearen lehenengo bideoa "K3-10.jpg" eta azkena "K3-561.jpg" da.

Ondoren, 4.9 Taulan, sortu den ARFF -aren laburpena ageri da. Erlazioaren izena, atributuak eta datuak ageri dira. Azkenengoan, klase bakoitzaren hasierako eta amaierako irudiak ageri dira soilik.

```
@relation GrAL
@attribute filename string
@attribute class K1, K2, K3
@data
K1-10.jpg, K1
K1-11.jpg, K1
...
K1-224.jpg, K1
K1-225.jpg, K1
K2-10.jpg, K2
K2-11.jpg, K2
...
K2-224.jpg, K2
K2-225.jpg, K2
K3-10.jpg, K3
K3-11.jpg, K3
...
K3-560.jpg, K3
K3-561.jpg, K3
```

4.9 Taula: ARFF adibidea

4.3.2 Iragezkiak aplikatu

Behin gure ARFF-a sortuta dagoenean, WEKA erabiltzen hasiko gara. Pausu hau aurre-prozesaketari dagokio. Fitxategi honi, zenbait iragezki aplikatuko zaizkio emaitza gehiago kalkulatu ahal izateko. ImageFilters zerrendako (3.20 Irudia) ondorengo lau iragezkiak aukeratu dira: AutoColor, Binary, ColorLayout eta PHOG. Oso garrantzitsua da iragezki bakoitza aplikatu ondoren filename izeneko atributua ezabatzea.

Atal hau bi modutan garatu daiteke, bai interfazearen bidez eta bai terminalaren bidez. Terminalaren bidez egiteko komandoak ondorengoak dira:

AutoColor iragezkiak:

```
java -cp weka.jar weka.Run
weka.filters.unsupervised.instance.imagefilter.AutoColorCorrelogramFilter -i irteera/ImageFilter_DB19.arff -D irteera/ordenatuak -o image_filter/AutoColor.arff
```

Binary iragezkiak:

```
java -cp weka.jar weka.Run
weka.filters.unsupervised.instance.imagefilter.BinaryPatternsPyramid -i irteera/ImageFilter_DB19.arff -D irteera/ordenatuak -o image_filter/Binary.arff
```

ColorLayout iragezkiak:

```
java -cp weka.jar weka.Run
weka.filters.unsupervised.instance.imagefilter.ColorLayoutFilter -i irteera/ImageFilter_DB19.arff -D irteera/ordenatuak -o image_filter/ColorLayout.arff
```

PHOG iragezkiak:

```
java -cp weka.jar weka.Run weka.filters.unsupervised.instance.imagefilter.PHOGFilter
-i irteera/ImageFilter_DB19.arff -D irteera/ordenatuak -o image_filter/PHOG.arff
```

Behin iragezkiak aplikatu diren, ARFF fitxategi bakoitzean atributu gisa izendatutako filename ezabatuko da. Kasu honetan ere, bai interfazearen bidez eta bai terminala erabiliz burutu daiteke. Kasu honetan, ez da script bat sortu atal hau burutzeko, ez baitira ARFF asko sortu behar. Hala ere, terminalaren bidez garatu da, hurrengo komandoen bidez:

```
java -cp weka.jar weka.Run weka.filters.unsupervised.attribute.RemoveType -T string
-i image_filter/AutoColor.arff -o image_filter/AutoColor.arff
```

```
java -cp weka.jar weka.Run weka.filters.unsupervised.attribute.RemoveType -T string
-i image_filter/Binary.arff -o image_filter/Binary.arff
```

```
java -cp weka.jar weka.Run weka.filters.unsupervised.attribute.RemoveType -T string
-i image_filter/ColorLayout.arff -o image_filter/ColorLayout.arff
```

```
java -cp weka.jar weka.Run weka.filters.unsupervised.attribute.RemoveType -T string
-i image_filter/PHOG.arff -o image_filter/PHOG.arff
```

4.3.3 Sailkapen sinplea

Iragazkiak aplikatu ondoren, sailkapenarekin hasteko ordua da. Guztira, 4 ImageFilter eta 17 sailkatzaile ditugu. Ondorioz, 68 emaitza aterako dira. Lana errazteko helburuarekin programatxo bat sortu da lan guzti hau automatizatzeko.

Script honetan, "*nire_script1.sh*" izenekoa, [3.6.3](#) eta [3.6.4](#) ataletako sailkatzaileak aplikatu dira bertan agertzen diren komandoekin. Lehenik eta behin, "*emaitzak*" izeneko karpeta sortuko du, emaitzak bertan gorde daitezten. Ondoren, loop bikoitza inplementatu da, iragazki gutiei sailkatzaile guztiak aplikatzeko helburuarekin. Honen laguntzaz lana erraztu da, denbora murriztuz.

Programa abiarazteko komandoa:

```
sh programak/nire_script1.sh
```

Script-a exekutatu ostean, "*emaitzak/P1_emaitzak*" izeneko direktorioan, arff bakoitzak sailkatzaile guztiekin lortu duen fitxategiak gorde dira. Gainera, informazio guzti hau laburtuz, "*emaitzak*" direktorioan ere, fitxategi solte bat sortu da asmatze-tasa guztiak biltzen dituen.

Adibidez, aipatutako fitxategiaren zati hau hartuz, ondorengo ondorioztu daiteke: AutoColor ARFF-ak J48 sailkatzailea aplikatuz, 568 instantzia zuzen sailkatu ditu, hau da, guztien % 57,7236. Emaitza guztiak [5.1](#) Taulan aurkitzen dira.

PHOG 7-NN		
Correctly Classified Instances	716	72.7642%
Correctly Classified Instances	568	54,0640%

4.3.4 Sailkapen konplexua

Orokorrean emaitza altuagoak espero zirenez, bigarren esperimentua egitea erabaki da, lehenengo esperimentuaren antzekoa izanik. Kasu honetan, sailkatzaile berdinak aplikatuko dira arff berdinei, baina aurretik atributu aukeraketa egingo da. "nire_script2.sh" izeneko programak honetaz arduratuko da.

sh programak/nire_script2.sh

Atributu aukeraketa, beste sailkatzaile bat eskatzen duen sailkatzailea bezala azaldu dezakegu. Ondorioz, aurreko esperimentuko sailkatzaileak alde batera utziz, berriak sortu beharko dira.

Kasu guztietan, sailkatzaile nagusia AttributeSelectedClassifier izango da eta "meta" karpeta aurkitzen da. Haren ezarpenetan hiru aukeraketa egin beharko dira. Alde bateatik, Evaluator atalean "ReliefAttributeEval" aukeratuko dugu eta bestetik, search atalean "Ranker". Kasu honetan, numToSelect -en 30 balioa sartuko dugu, algoritmoak 30 atributu aukera ditzen. Amaitzeko, classifier atalean lehenengo esperimentua balitz bezala, nahi dugun sailkatzailearen aukeraketa egingo dugu.

Esperimentu honetan, aurrekoarekin alderatuz, emaitza hobetu dela ikusi dezakegu, % 4,1677 zehazki. Kasu honetan, % 58,2317-ko asmatze tasa lortu da, sailkatzaile eta ARFF berdinarekin. Emaitza guztiak [5.2](#) taulan aurkitzen dira.

PHOG 7-NN		
Correctly Classified Instances	716	72.7642%
Correctly Classified Instances	568	58.2317%

4.3.5 Bozketa

Esperimentuko azkenengo atalean Bozketa izeneko prozesua burutuko da, 5.1 eta 5.2 Taulako datuak erabiliz. Teknika hau emaitza zehatzagoak lortzeko helburuarekin garatu da. Orain arte, sailkatzaile independenteak aplikatu dira. Hemendik aurrera, emaitzak aztertuz eta amankomunean jarriz, emaitza desberdinak lortu dira.

Lehenik eta behin, ARFF bakoitzetik hiru asmatze-tasa altuenak atera dituzten sailkatzaileak aukeratu behar dira. Kasu batzuetan, berdinketa sortzen da eta hiru baino sailkatzaile gehiago lortzen dira. Hori ekiditeko, horietako bat ausaz baztertzen da guztira hiru izateko.

5.2 Taulako datuak aztertuz, hauek dira sailkatzaile onenak ARFF bakoitzeko:

	AutoColor
KNN7	% 58,9431
KNN11	% 59,0447
RF	% 59,7561

4.10 Taula: AutoColor.arff emaitzarik altuenak

	Binary
RF	% 58,1301
SMO	% 56,0976
bag_J48	% 56,0976

4.11 Taula: Binary.arff emaitzarik altuenak

	ColorLayout
KNN1	% 64,3293
KNN3	% 64,8374
RF	% 67,6929

4.12 Taula: ColorLayout.arff emaitzarik altuenak

	PHOG
KNN1	% 64,8374
RF	% 60,1626
stack_SMO	% 59,8577

4.13 Taula: PHOG.arff emaitzarik altuenak

Ondoren, [4.10](#), [4.11](#), [4.12](#) eta [4.13](#) Taulako sailkatzailerekin lan egingo da, beste guztiak baztertuz. Horretarako, sortu dituzten emaitzak gordeko dira, baina soilik "Prediction on test data" ataleko datuak. Garrantzitsua da datu hauek CSV formatuan eskatzea, haiekin modu errezean lan egiteko.

[4.14](#) Taulako datuak adibide gisa izanik, ondorengoa ondorioztatzen da: lehenengo instantzia 3:K3 klasekoa da eta 3:K3 klasekoa dela aurreikusi da, 0,61-ko segurtasunarekin. Bigarren instantzia 2:K2 klasekoa da eta 3:K3 klasekoa dela aurreikusi da 0,427-ko segurtasunarekin. Hirugarren instantzia 3:K3 klasekoa da eta 2:K2 klasekoa dela aurreikusi da 0,533-ko segurtasunarekin.

inst,actual,predicted,error,prediction
1,3:K3,3:K3,,0.61
2,2:K2,1:K1,+,0.427
3,3:K3,2:K2,+,0.533

4.14 Taula: Bozketa adibidea

Bozketarekin amaitzeko, CSV fitxategi hauekin lan egingo dugu. Instantzia bakoitzeko hiru sailkapen izanik, ARFF multzo bakoitzean ondorengoa kalkulatu da: hiru kasuetatik zenbat alditan asmatu den benetako klasea, eta gutxienez bi boto alde lortu badira, ontzat eman da sailkapenaren prozesua.

Amaieran, asmatze-tasa orokorra lortuko da, [5.5](#) Taulan bilduta dagoena.

5. KAPITULUA

Emaitzak

Atal honetan, proiektuaren garapenean zehar lortu diren emaitzak bildu dira. Hori ez ezik, haien harira egin diren ondorioak eta hausnarketak ere gehitu dira.

5.1 Sailkapen sinplea

Lehenengo esperimentukoko emaitzak aztertuz zenbait ondorio atera dira.

- Emaitzarik altuena lortu duen ARFF-a Autocolor.arff izan da, % 59,0157 -ko asmatze-tasarekin.
- Emaitzarik baxuena lortu duen ARFF-a PHOG.arff izan da, % 49,7010 -ko asmatze-tasarekin.
- Emaitzik altuena lortu duen sailkatzailea 1-NN izan da ColorLayout.arff aztertzerakoan, % 64,939 -ko asmatze-tasarekin.
- Emaitzarik baxuena lortu duen sailkatzailea Naive Bayes izan da ColorLayout.arff aztertzerakoan, % 40,442 -ko asmatze-tasarekin.
- Naive Bayes erabiltzen duten sailkatzaileak hiru kasutan ez dute emaitzarik lortu. Hirurak AutoColor.arff -an gertatu dira, Naive Bayes, Bagging eta Boosting prozesuetan zehazki.

- K-NN sailkatzailean, K -ren balioa handitu ahala, asmatze-tasa ere handitu da.
- Bagging, boosting eta stacking teknikak, orokorrean, sailkatzaile sinpleen emaitzak hobetu dituzte. Beste kasutan, emaitza berdina lortu da, hala nola, SMO eta boost_SMO sailkatzaileekin.

	AutoColor	Binary	ColorLayout	PHOG
J48	% 57,7236	% 48,1707	% 58,6382	% 45,3252
KNN1	% 58,9431	% 50,6098	% 64,939	% 53,4553
KNN3	% 61,5854	% 50,9146	% 64,126	% 51,2185
KNN7	% 61,1789	% 52,3374	% 61,5854	% 54,064
KNN11	% 61,1789	% 52,7439	% 61,687	% 55,3862
NB	-	% 46,748	% 40,442	% 43,2927
RF	% 59,4512	% 48,374	% 57,622	% 41,7683
SMO	% 58,3333	% 55,9959	% 59,4512	% 49,3902
bag_J48	% 60,3659	% 50,9146	% 62,3984	% 52,7439
bag_NB	-	% 47,561	% 41,4634	% 42,9872
bag_SMO	% 58,5366	% 54,3699	% 59,4512	% 51,5244
boost_J48	% 57,8252	% 51,4228	% 60,061	% 51,5244
boost_NB	-	% 46,748	% 41,4634	% 43,2927
boost_SMO	% 58,3333	% 51,4228	% 59,4512	% 47,9675
stack_5NN	% 55,6911	% 46,9512	% 60,7724	% 51,5244
stack_J48	% 56,8089	% 56,8089	% 61,5854	% 53,4553
stack_SMO	% 60,2642	% 56,4024	% 60,7724	% 56,0976
Batezbestekoa	% 59,0157	% 50,9625	% 57,4722	% 49,7010

5.1 Taula: Sailkapen sinplearen emaitzak

5.2 Sailkapen konplexua

Bigarren esperimentuko emaitzak aztertuz zenbait ondorio atera dira.

- Emaizarik altuena lortu duen ARFF-a ColorLayout.arff izan da, % 58,0290 -ko asmatze-tasarekin.
- Emaizarik baxuena lortu duen ARFF-a PHOG.arff izan da, % 54,4636 -ko asmatze-tasarekin.
- Emaizik altuena lortu duen sailkatzailea RF izan da ColorLayout.arff aztertzerakoan, % 67,6929 -ko asmatze-tasarekin.
- Emaizarik baxuena lortu duen sailkatzaileak Naive Bayes eta boost_NB izan dira ColorLayout.arff aztertzerakoan, % 40,8537 -ko asmatze-tasarekin.
- RF sailkatzaileak lau kasutan hiru emaitzarik altuenean kokatu da.
- Aurreko esperimentuan ez bezala, K-NN algorimoak, K-ren hazkuntza dela eta, ez ditu beti emaitzak hobetu.
- RF sailkatzaileak lau kasutan hiru emaitzarik altuenean kokatu da.

	AutoColor	Binary	ColorLayout	PHOG
J48	% 57,7236	% 50,3049	% 58,0285	% 47,9675
KNN1	% 57,4187	% 50,813	% 64,3293	% 64,8374
KNN3	% 58,3333	% 49,4919	% 64,8374	% 57,8252
KNN7	% 58,9431	% 51,626	% 63,313	% 58,2317
KNN11	% 59,0447	% 53,5569	% 63,4146	% 57,5203
NB	% 55,7327	% 46,0366	% 40,8537	% 46,6463
RF	% 59,7561	% 58,1301	% 67,6929	% 60,1626
SMO	% 58,435	% 56,0976	% 59,0447	% 56,0976
bag_J48	% 57,9268	% 56,0976	% 62,5000	% 54,5732
bag_NB	% 55,5894	% 44,9187	% 42,0732	% 46,1382
bag_SMO	% 58,1201	% 56,0976	% 60,1626	% 56,1992
boost_J48	% 57,7236	% 52,1341	% 56,8089	% 51,8293
boost_NB	% 55,7327	% 46,0366	% 40,8537	% 46,6463
boost_SMO	% 58,435	% 56,0976	% 59,0447	% 55,5559
stack_5NN	% 54,9797	% 47,4593	% 60,6707	% 52,5407
stack_J48	% 57,1138	% 52,0325	% 59,8577	% 53,252
stack_SMO	% 58,8415	% 50,3008	% 63,0081	% 59,8577
Batezbestekoa	% 57,638	% 51,6019	% 58,0290	% 54,4636

5.2 Taula: Sailkapen konplexuaren emaitzak

5.3 Alderaketa

Bi esperimentuen arteko emaitzak aztertuz zenbait ondorio atera dira.

- Hobekuntza altuena lortu duen ARFF-a PHOG.arff izan da, % 4,7625 -ko asmatze-tasa handituz.
- Hobekuntza baxuena lortu duen ARFF-a AutoColor.arff izan da, % 1,3775 -ko asmatze-tasa txikituz.
- Hobekuntza altuena lortu duen sailkatzailea RF izan da, % 18,3943 -ko asmatze-tasa handituz.
- Hobekuntza baxuena lortu duen sailkatzailea stack_SMO izan da, % 6,1016 -ko asmatze-tasa txikituz.
- PHOG.arff -an kasu guztietan emaitza hobetu da, stack_J48 sailkatzailean izan ezik.
- RF sailkatzaileak lau ARFF-etan emaitzak hobetu ditu.
- J48 sailkatzaileak emaitza berdina aurreikusi du AutoColor ARFF-an, hobekuntza 0 ezarriz.

	AutoColor	Binary	ColorLayout	PHOG
J48	% 0	% 2,1342	% -0,6097	% 2,6423
KNN1	% -1,5244	% 0,2032	% -0,6097	% 11,3821
KNN3	% -3,2521	% -1,4227	% 0,7114	% 6,6067
KNN7	% -2,2358	% -0,7114	% 1,7276	% 4,1677
KNN11	% -2,1342	% 0,813	% 1,7276	% 2,2341
NB	-	% -0,7114	% 0,4117	% 3,3536
RF	% 0,3049	% 9,7561	% 10,0709	% 18,3943
SMO	% 0,1017	% 0,1017	% -0,4065	% 6,7074
bag_J48	% -2,4391	% 5,183	% 0,1016	% 1,8293
bag_NB	-	% -2,6423	% 0,6098	% 3,151
bag_SMO	% -0,4165	% 1,7277	% -0,4065	% 4,6748
boost_J48	% -0,1016	% 0,7113	% -3,2521	% 0,3049
boost_NB	-	% -0,7114	% -0,6097	% 3,3536
boost_SMO	% 0,1017	% 4,2683	% -0,4065	% 7,5884
stack_5NN	% -0,7114	% 0,5081	% -0,1017	% 1,0163
stack_J48	% 0,3049	% -2,2358	% -1,7277	% -0,2033
stack_SMO	% -1,4227	% -6,1016	% 2,2357	% 3,7601
Batezbestekoa	% -1,3775	% 0,6394	% 0,5568	% 4,7625

5.3 Taula: Saikapen sinplearen eta konplexuaren arteko desberdintasuna

5.4 Bozketa

- Kasu guztietan emaitzak hobetu dira, batezbestekoarekin alderatuta.
- Hazkunde nabariena ColorLayout ARFF-an gertatu da, sailkapen konplexuan. % 11,6867 -ko hobekuntza lortu du.
- Hazkunde txikiena AutoColor ARFF-an gertatu da, sailkapen sinplean. % 2,0615 -ko hobekuntza lortu du.

	AutoColor	Binary	ColorLayout	PHOG
Batezbestekoa	% 59,0157	% 50,9625	% 57,4722	% 49,7010
Bozketa	% 61,0772	% 56,0976	% 65,6504	% 56,0976

5.4 Taula: Sailkapen sinplearen bozketa

	AutoColor	Binary	ColorLayout	PHOG
Batezbestekoa	% 57,638	% 51,6019	% 58,0290	% 54,4636
Bozketa	% 61,8902	% 60,6707	% 69,7154	% 64,1260

5.5 Taula: Sailkapen konplexuaren bozketa

6. KAPITULUA

Ondorioak eta etorkizuneko lana

Proiektu guztietan bezala, honetan ere, ateratako ondorioak hausnartuko dira atal honetan. Gainera, proiektu honekin lanean jarraitzeko zenbait aukera aztertuko dira, etorkizuneko lanean garatzen jarraitzeko asmoarekin.

6.1 Ondorioak

Azpiatal honetan proiektuaren bukaeran lortutako emaitzak aztertu dira eta haien analisisa gauzatu da. Honekin batera, proiektua aurrera eramateari esker, lortu den ikasketa pertsonala azaltzen da.

Hasteko, lortutako emaitzak aztertuz honako hau ondorioztatu da. Irudiak sailkatzeko sailkatzaile asko erabili direnez, emaitza ugari lortu dira, batzuk oso onak izan direnak, eta beste batzuk, ez hainbeste. Ondorioz, garrantzitsua da sailkatzaile bakoitzaren erabilera zuzena ezagutzea. Sailkapen konplexuan, sinplean baino emaitza hobeak lortu dira. Nahiz eta kasu gehienetan aldea ez den oso nabaria izan, hobekuntza argi lortu da. Hortaz, multisailkatzaileek lan hobea egin dutela esan daiteke.

Garapenaren azkenengo atalean, sailkapen sinplea egiterakoan lortutako emaitzak ez dira espero bezalakoak izan. Hori dela eta, bigarren sailkapena egitea erabaki izan da emaitzak alderatzeko. Gainera, bozketa eginez lortu diren emaitzak onak izan direnez, ase sentitu naiz egindako lanarekin.

Lanari dagokionez, Informatika Ingeniaritzako Gradu osoan zehar burutu den lanik zorrotzena izan da, ordu gehien eskatzen duen ikasgaia baita. Hori dela eta, ezinbestekoa izan

da lanaren jardunean egunerokotasuna mantentzea. Nahiz eta proiektua COVID-19aren pandemiaren erdian burutu izan den, normaltasun osoz lan egin dela baieztatu daiteke. Hala ere, ezohiko egoera batera moldatzera behartuak izan gara zuzendariekin harremanetan jartzeko orduan, lanari ikuspuntu errealista eskainiz.

Eta ezagutzari dagokionez, mota askotako kontzeptuak ikasi dira proiektua garatu bitartean. Alde batetik, beste lanetan erabilitako softwareen erabilera zabaldu da, tresna berriak erabiliz. Bestetik, inoiz landu diren baliabideak erabiltzen ikasi dira, eta modu berean, graduan zehar ikasitako ezagutzak errepasatzeko aukera egon da. Ondorioz, beharrezkoa izan da erabilitako software eta sistemen ezagutza egokiak izatea, behar den egoerara moldatzeko aukera izateko.

6.2 Etorkizuneko lana

Proiektu gehienetan bezala, honetan ere badago luzatzeko aukera, teknika berriak aplikatuz edota proba berriak eginez. Ondoren etorkizunerako proposatzen diren proposamenak zerrendatzen dira:

1. Proiektu honetan hiru klaseetako datu-basea erabili da. Honen ordean, datu-base berri eta handiagoa bilatu daiteke, klase gehiagoz osatuta dagoena. Honekin emaitza desberdinak aterako direla argi dago.
2. Fluxu optikoa kalkulatzeko indarra aukeratu behar denez, balio desberdinak probatu daitezke. Hau eginez, irudi desberdinak sortuko lirateke.
3. Aurreikuspenaren aurreprozesaketari dagokionez, iragazki desberdinak aplikatzea interesgarria izango litzateke. Modu honetan, irudien ezaugarri desberdinak nabarmentzea lortuko lirateke.
4. Era berean, sailkatzaile desberdinak aplikatzea emaitza anitz gehiago ateratzeko oso aproposa da.
5. Amaitzeko, bozketari dagokionez, hiru sailkatzaile onenak aukeratu beharrean, bost onenak aukeratuz emaitza zehatzagoak aterako lirateke.

Hurrengo urteari begira, posible bada, lan mota hau bezalako proiektua egitea gustatuko litzaidake, dagokion aurrerapenak eta aldaketak eginez.

7. KAPITULUA

Eranskina: Kodeak

- p1_bideoak_zatitu.sh

```
# Argumentuak gorde
input=$1
output=$2
sframe=$3
eframe=$4
# Zatitu hasiera denbora lortzeko
stime='echo $sframe/25|bc -l'
# 0 segunduan hasten bada 0 zifra gehitu
if [ 'echo $stime|head -c 1' = "." ]; then
stime="0$stime"
fi
# Zatitu bukatze denbora lortzeko
etime='echo $eframe/25|bc -llhead -c 5'
# 0 segunduan bukatzen bada 0 zifra gehitu
if [ 'echo $etime|head -c 1' = "." ]; then
etime="0$etime"
fi
# Bideoaren iraupena
dur='echo $etime-$stime|bc -l'
1 segundu baino gutxiago bada 0 zifra gehitu
if [ 'echo $dur|head -c 1' = "." ]; then
dur="0$dur"
fi
# Komandoa
ffmpeg -i $input -ss $stime -t $dur -async 1 -c copy $output
```

-
- p2_optflow2.m

```
clear all; close all;
global path opticFlow
path = './DBak/BOSS/';
opticFlow = opticalFlowHS('Smoothness',3,'MaxIteration',30,'VelocityDifference',0);
klaseak = ["sitting","standing","walking"];
ezabatu_eta_sortu(append(path,'csv/'));
listing = dir(path);
for i = 1:length(listing)
if any(strcmp(klaseak,listing(i).name))
path_class = append(path,'csv/',listing(i).name,');
bideoak = dir(fullfile(append(path,listing(i).name), '*.avi'));
for j = 1:length(bideoak) [ ,bideo_izena, ] = fileparts(bideoak(j).name);
mkdir(append(path_class,bideo_izena));
optflow_fun(listing(i).name,bideo_izena,path_class,'vx')
optflow_fun(listing(i).name,bideo_izena,path_class,'vy')
optflow_fun(listing(i).name,bideo_izena,path_class,'mg')
end;end;end
```

```
function optflow_fun(class,video,path_class,ald)
global path opticFlow
path_ald = append(path_class,video,'/',video,'_',ald);
mkdir(path_ald);
v = VideoReader(append(path,class,'/',video,'.avi'));
j = 0;
while hasFrame(v)
frameRGB = readFrame(v);
frameGray = imresize(rgb2gray(frameRGB), 0.25);
flow = estimateFlow(opticFlow,frameGray);
path_math = append(path_ald,"/",video,'_',ald,'_f',int2str(j),'.csv');
if strcmp(ald,'vx')
csvwrite(path_math, flow.Vx)
elseif strcmp(ald,'vy')

csvwrite(path_math, flow.Vy)
elseif strcmp(ald,'mg')
csvwrite(path_math, flow.Magnitude)
end
j=j+1;
end
end
function ezabatu_eta_sortu(path)
if exist(path,'dir')
rmdir(path,'s');
end
mkdir(path);
end
```

- p3_aldagaiak2.m

```
clear all; close all;
global lf cf
%% Hasieraketak
lf = 144; cf = 180;
path = './DBak/BOSS/csv/';
clases = ["sitting", "standing", "walking"];
ezabatu_eta_sortu('./DBak/BOSS/csv_aldagaiak');
%% Karpeta horren fitxategiak
listing = dir(path);
for i = 1:length(listing)
    % Karpeta klase baten izena da
    if any(strcmp(clases,listing(i).name))
        path_class = append(path,listing(i).name,"/");
        bideoak = dir(fullfile(path_class));
        % Bideo guztien karpetak zeharkatu
        for j = 1:length(bideoak)
            if (strcmp('.',bideoak(j).name) || strcmp('..',bideoak(j).name))
                path_video = append(path_class,bideoak(j).name,"/");
                delete(append(path_video,'*.csv'))
                aldagaiak = dir(fullfile(path_video));
                % Bideo bakoitzaren hiru aldagaiak zeharkatu
                for t = 1:length(aldagaiak)
                    if (strcmp('.',aldagaiak(t).name) || strcmp('..',aldagaiak(t).name))
                        m = complete_matrix(path_video,aldagaiak(t).name);
                        calculate(m,'./DBak/BOSS/csv_aldagaiak/',aldagaiak(t).name);
                    end;end;end;end;end;end;end
```

```

%% Aldagai bakoitzeko matrize guztiak elkartuko ditu
function matrix = complete_matrix(path_video,ald)
global lf cf
% Bideo - aldagai bakoitzaren csvak
path_ald = append(path_video,ald);
csvak = dir(fullfile(path_ald,"*.csv"));
% Zenbat csv
s = length(csvak);
% Matrize hutsa
matrix = zeros(lf,cf,s);
% csv guztiak zeharkatu
for c = 1:s
mycsv = csvread(append(path_ald,"/",csvak(c).name));
% Pone todas las matrices de la misma variable una detrás de otra
matrix(:,:,c) = mycsv;
end;end
%% Calcular media, varianza y kurtosis
function calculate(matrix,path1,path2)
global lf cf
% Matrices vacías
bb_csv = zeros(lf,cf); bar_csv = zeros(lf,cf);
kur_csv = zeros(lf,cf);
for y = drange(1:lf)
for x = drange(1:cf)
% Todos los datos del mismo pixel
aux = matrix(y,x,:);
bb_csv(y,x) = mean(aux); bar_csv(y,x) = var(aux); kur_csv(y,x) = kurtosis(aux);
end;end
% Guardar matrices
csvwrite(append(path1,path2,"_bzb.csv"),bb_csv)
csvwrite(append(path1,path2,"_bar.csv"),bar_csv)
csvwrite(append(path1,path2,"_kur.csv"),kur_csv)
end

function ezabatu_eta_sortu(path)
if exist(path,'dir')
rmdir(path,'s');
end
mkdir(path);
end

```

- p4_normalize4.m

```
clear all; close all;
global klaseak
path = './DBak/BOSS/csv_aldagaiak/';
path_img = './DBak/BOSS/rgb_irudiak/';
klaseak = ["sitting", "standing", "walking"];
ezabatu_eta_sortu(path_img);
csvak = dir(fullfile(path, '*.csv'));
n = size(csvak); n = n(1); i = 1;
while i <= n
    % Hirunaka hartu eta normalizatu
    data_bb_scaled = normalize(csvread(append(path, csvak(i+1).name)));
    data_bar_scaled = normalize(csvread(append(path, csvak(i).name)));
    data_kur_scaled = normalize(csvread(append(path, csvak(i+2).name)));
    % Matrizatu hutsa
    data_rgb = zeros(144, 180, 3);
    % Matrizatu berria osatu
    data_rgb(:, :, 1) = data_bb_scaled;
    data_rgb(:, :, 2) = data_bar_scaled;
    data_rgb(:, :, 3) = data_kur_scaled;
    irudia = uint8(data_rgb);
    if strcmp('sit', csvak(i).name(1:3))
        kl = 'sitting/';
    elseif strcmp('sta', csvak(i).name(1:3))
        kl = 'standing/';
    elseif strcmp('wal', csvak(i).name(1:3))
        kl = 'walking/';
    end
    % Irudiak gorde
    imwrite(irudia(:, :, 1), append(path_img, kl, csvak(i).name(1:18), '_r.jpg'), 'jpg');
    imwrite(irudia(:, :, 2), append(path_img, kl, csvak(i).name(1:18), '_g.jpg'), 'jpg');
    imwrite(irudia(:, :, 3), append(path_img, kl, csvak(i).name(1:18), '_b.jpg'), 'jpg');
    imwrite(irudia, append(path_img, kl, csvak(i).name(1:18), '_rgb.jpg'), 'jpg');
    i = i+3;
end
```

```
% Funtzio lagungarria: Datuak normalizatu
function s = normalize(d)
minData = min(d(:));
maxData = max(d(:));
s = ((d - minData) / (maxData - minData))*255; % Scaled to [0, 255]
s = floor(s); % borobildu gora
end

%% Funtzio lagungarria
function ezabatu_eta_sortu(path)
global klaseak
if exist(path,'dir')
rmdir(path,'s');
end

mkdir(path);
for i = 1:length(klaseak)
mkdir(append(path,klaseak(i)))
end
end
```

Bibliografia

- [1] Schonfeld, D., Shan, C., Tao, D., and Wang, L. (2010). Videosearch and mining, volume 287. Springer.
- [2] Kong, Y. and Fu, Y. (2018). Human action recognition and prediction: A survey. arXiv preprint arXiv:1806.11230.
- [3] Robotika eta Sistema Autonomoen Ikerketa taldea (RSAIT). <https://www.ehu.eus/es/web/enpresa/-/rsait>
- [4] BOSS datu-basea. <http://velastin.dynu.com/videodatasets/BOSSdata/>
- [5] Fluxu optikoa. https://en.wikipedia.org/wiki/Optical_flow
- [6] Weka 3: Machine Learning Software in Java. <https://www.cs.waikato.ac.nz/ml/weka/>
- [7] MATLAB. Help Center. <https://www.mathworks.com/help/>
- [8] Comma-separated values. https://en.wikipedia.org/wiki/Comma-separated_values
- [9] FFmpeg. <https://www.ffmpeg.org/>
- [10] ImageFilter. <https://github.com/mmayo888/ImageFilter/master/weka/filters/unsupervised/instance/imagefilter>
- [11] Interface Classifier. <https://weka.sourceforge.io/doc.dev/weka/classifiers/Classifier.html>
- [12] Método de Horn–Schunck. https://es.wikipedia.org/wiki/M%C3%A9todo_de_Horn%E2%80%93Schunck