

# ГАРЫН ХЭЛБЭР ТАНИХ

Hand Gesture Recognition

# ЗОРИЛГО

Камераар дамжуулан гарын ямар хэлбэр гаргасан буюу хэдэн хуруу гаргасныг машин сургалтын аргаар таньж, түүнийгээ хялбар тоглоомонд ашиглах.

# ТАНИХ ҮЙЛ АЖИЛЛАГАА

1. Зураг хүлээж авах
2. Зураг боловсруулах
3. Шинж чанаруудыг ялгаж авах
4. Таних
5. Шийдвэр гаргах

# АШИГЛАСАН ТЕХНОЛОГИ

- Python
- OpenCV (Open Source Computer Vision Library)
- scikit-learn (Machine Learning in Python)
- Numpy (Fundamental package for scientific computing with Python)
- PyAutoGUI (GUI automation for human beings)

# АШИГЛАСАН ТЕХНОЛОГИ

1. Зураг хүлээж авах - Камер
2. Зураг боловсруулах - OpenCV
3. Шинж чанаруудыг ялгаж авах - OpenCV
4. Таних - scikit-learn SVM
5. Шийдвэр гаргах - Тоглоом/PyAutoGUI

# ЗУРАГ ХҮЛЭЭЖ АВАХ

Камераар хүлээн авч буй зургуудыг ашиглах бөгөөд танилтыг сайжруулахын тулд гар нь орчноос өөр өнгөтэй байх хэрэгтэй.

```
# OpenCV
import cv2

cap = cv2.VideoCapture(0)
while True:
    ret, img = cap.read()
```

Камернаас зураг унших (OpenCV)



# ЗУРАГ БОЛОВСРУУЛАХ



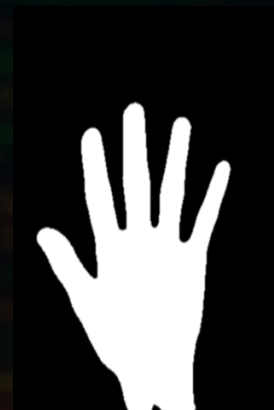
Арын  
фон



Үндсэн  
зураг



Арын  
фоныг  
хассан

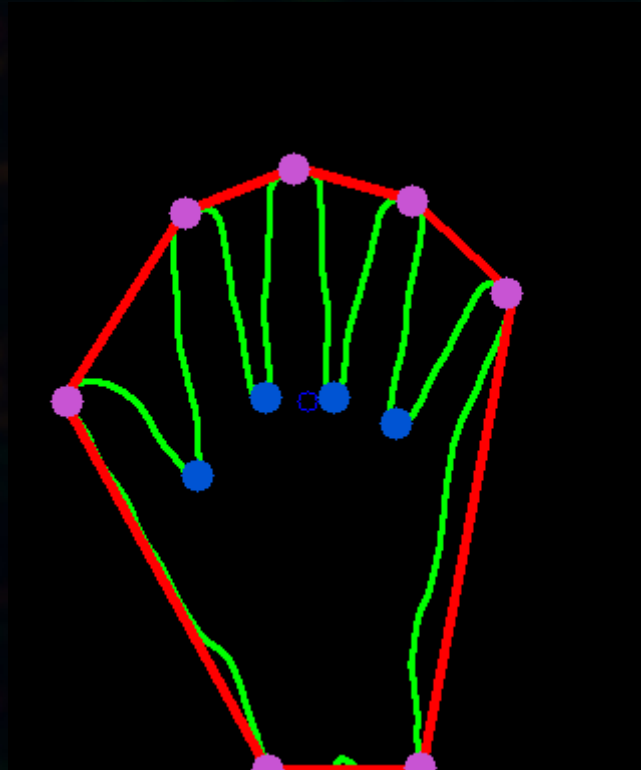


Зааглах

```
# Background
img = cv2.absdiff(bgImg, image)
# Gray
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# Gaussian Blur
blur = cv2.GaussianBlur(gray, (11, 11), 0)
# Threshold
ret, thresh = cv2.threshold(blur, threshold, 255, cv2.THRESH_B
# Gaussian Blur
thresh = cv2.GaussianBlur(thresh, (3, 3), 0)
```



# ШИНЖ ЧАНАРУУДЫГ ЯЛГАХ



---

Гарын өнцгүүд дээрх  $X, Y$  координатуудыг олох

---

Countours | Convex hull | Convexity defects

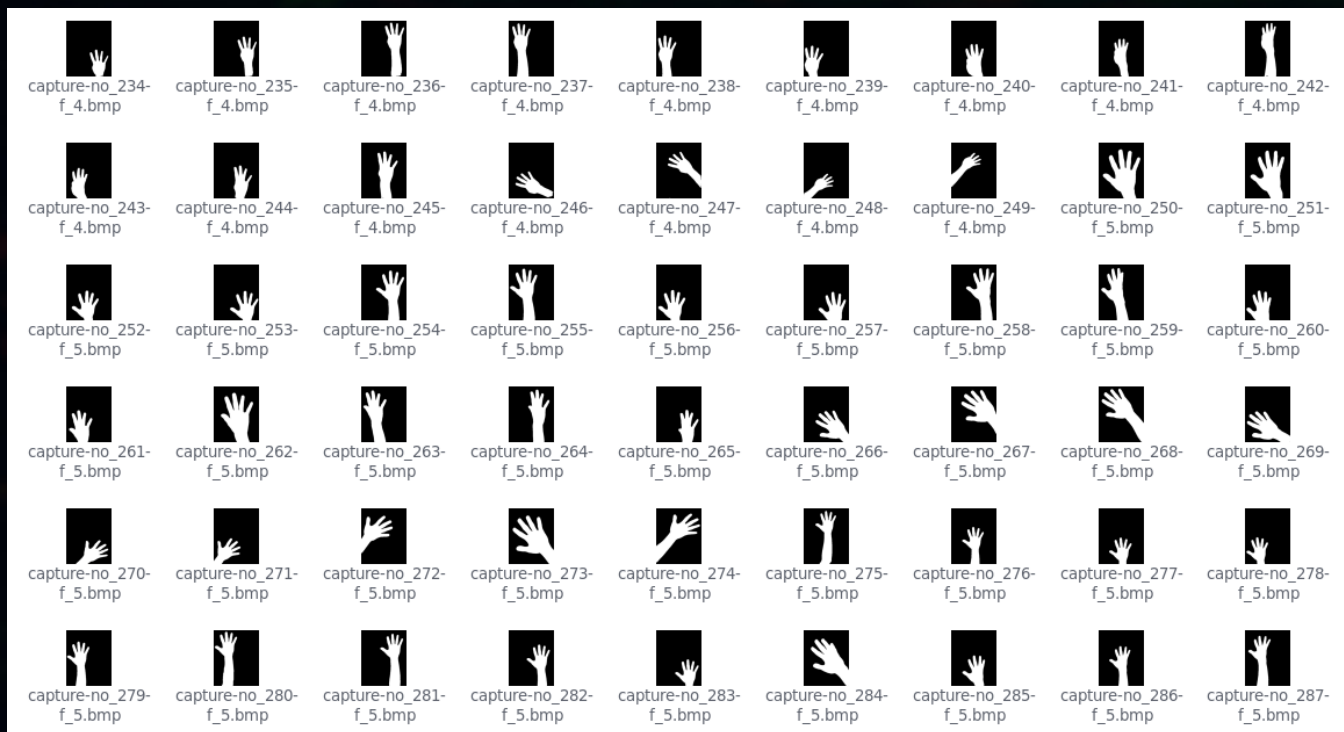
```
cv2.findContours()
```

```
cv2.convexHull()
```

```
cv2.convexityDefects()
```

# ТАНИХ

Зурагнаас олж буй шинж чанаруудыг гарын  
хөдөлгөөн бүрт хадгалан нийт 300 ширхэг жишээ  
өгөгдөл цуглуулсан.



№	label	x1	y1	x2	y2	x3	y3	x4	y4	x5	y5	x6	y6	x7	y7	a_x1	a_y1	a_x2	a_y2	a_x3	a_y3	a_x4	a_y4	filename
0	0	19	25	230	110	206	28	105	320	136	0	223	318	0	121	0	0	0	0	0	0	0	0	0 captures2/capture-no_0-f_0.bmp
1	0	88	317	56	0	11	144	172	31	0	28	173	317	158	17	0	0	0	0	0	0	0	0	0 captures2/capture-no_1-f_0.bmp
2	0	3	51	56	317	184	24	105	0	20	32	165	322	0	78	0	0	0	0	0	0	0	0	0 captures2/capture-no_2-f_0.bmp
3	0	126	0	261	296	7	134	269	174	53	20	179	42	0	62	0	0	0	0	0	0	0	0	0 captures2/capture-no_3-f_0.bmp
4	0	77	7	261	288	0	92	263	189	60	148	115	0	18	37	0	0	0	0	0	0	0	0	0 captures2/capture-no_4-f_0.bmp
5	0	243	176	50	0	0	127	239	293	175	73	10	42	90	4	0	0	0	0	0	0	0	0	0 captures2/capture-no_5-f_0.bmp
6	0	188	26	234	165	0	252	55	78	36	328	124	0	226	90	0	0	0	0	0	0	0	0	0 captures2/capture-no_6-f_0.bmp
7	0	257	90	0	281	166	0	224	22	119	48	76	294	254	121	0	0	0	0	0	0	0	0	0 captures2/capture-no_7-f_0.bmp
8	0	245	88	96	9	0	313	205	27	0	189	121	0	53	313	0	0	0	0	0	0	0	0	0 captures2/capture-no_8-f_0.bmp
9	0	0	76	139	34	150	185	36	0	88	0	1	25	66	188	0	0	0	0	0	0	0	0	0 captures2/capture-no_9-f_0.bmp
10	0	104	0	170	354	0	36	141	30	64	356	150	73	40	0	0	0	0	0	0	0	0	0	0 captures2/capture-no_10-f_0.bmp
11	0	20	5	135	26	48	149	0	64	126	148	92	0	141	76	0	0	0	0	0	0	0	0	0 captures2/capture-no_11-f_0.bmp
12	0	138	334	16	15	133	20	47	337	0	72	147	77	83	0	0	0	0	0	0	0	0	0	0 captures2/capture-no_12-f_0.bmp
13	0	57	0	170	349	141	38	68	353	3	23	0	68	107	6	0	0	0	0	0	0	0	0	0 captures2/capture-no_13-f_0.bmp
236	4	0	26	100	367	134	36	43	0	44	369	88	4	118	336	51	8	26	0	0	4	0	0 captures2/capture-no_236-f_4.bmp	
237	4	42	0	115	365	127	38	8	134	0	26	31	365	87	7	51	9	26	0	0	1	0	0 captures2/capture-no_237-f_4.bmp	
238	4	0	21	27	282	127	53	0	135	96	282	47	0	90	12	48	13	25	0	0	1	0	0 captures2/capture-no_238-f_4.bmp	
239	4	0	148	51	0	129	56	90	212	2	18	29	211	96	16	47	14	26	1	0	0	0	0 captures2/capture-no_239-f_4.bmp	
240	4	29	210	39	0	126	33	10	184	0	37	122	221	81	0	56	15	29	0	0	15	0	0 captures2/capture-no_240-f_4.bmp	
241	4	0	142	69	0	103	256	2	27	49	256	101	26	34	0	49	0	25	6	0	14	0	0 captures2/capture-no_241-f_4.bmp	
242	4	104	23	107	367	8	31	0	136	67	0	23	368	108	320	50	0	25	13	0	21	0	0 captures2/capture-no_242-f_4.bmp	
243	4	109	209	50	0	0	153	113	45	19	189	6	25	86	7	52	0	27	0	0	8	0	0 captures2/capture-no_243-f_4.bmp	
244	4	80	209	45	0	127	75	11	224	1	11	0	159	91	23	47	45	29	0	0	6	0	0 captures2/capture-no_244-f_4.bmp	
245	4	24	323	0	14	141	68	4	159	99	322	99	21	48	0	49	35	29	0	0	14	0	0 captures2/capture-no_245-f_4.bmp	
246	4	264	152	0	90	102	148	38	14	95	0	239	177	7	38	35	0	11	11	0	32	0	0 captures2/capture-no_246-f_4.bmp	
247	4	91	0	234	243	0	91	233	163	6	43	37	14	235	184	34	0	12	13	0	33	0	0 captures2/capture-no_247-f_4.bmp	
248	4	206	85	94	40	52	169	165	0	0	148	211	49	199	20	13	13	0	0	15	32	0	0 captures2/capture-no_248-f_4.bmp	
249	4	95	40	2	234	211	45	161	0	0	149	207	78	198	18	15	11	0	0	20	29	0	0 captures2/capture-no_249-f_4.bmp	
250	5	235	337	45	46	262	61	0	182	121	0	196	10	145	337	111	14	72	0	32	7	0	70 captures2/capture-no_250-f_5.bmp	
251	5	239	46	0	179	228	326	55	46	116	0	142	328	185	0	100	0	62	1	25	15	0	68 captures2/capture-no_251-f_5.bmp	
252	5	0	120	151	17	149	204	47	21	106	0	188	60	86	204	78	17	56	0	26	4	0	45 captures2/capture-no_252-f_5.bmp	
253	5	198	59	0	128	150	201	43	25	155	15	106	0	89	202	81	13	55	0	24	1	0	41 captures2/capture-no_253-f_5.bmp	
254	5	45	28	165	341	198	49	0	128	100	0	85	341	153	9	81	11	54	0	25	4	0	47 captures2/capture-no_254-f_5.bmp	

Цуглуулсан өгөгдлүүдээ SVM (Support vector machine) машин сургалтын аргыг ашиглан сургаж, танилтанд ашигласан.

```
# scikit-learn svm
from sklearn import svm

# Train data
tdata = [[...], [...], ...]
tlabel = [...]
clf = svm.SVC(kernel='linear', C=1.0)
clf.fit(tdata, tlabel)

# Predict
clf.predict([coords])
```

# ШИЙДВЭР ГАРГАХ

Зурагнаас таньсан дүрснээс хамаарч тоглоомонд  
зохих үйлдлүүдийг хийх



There is no Internet connection

Try:

- Checking the network cables, modem, and router
- Reconnecting to Wi-Fi

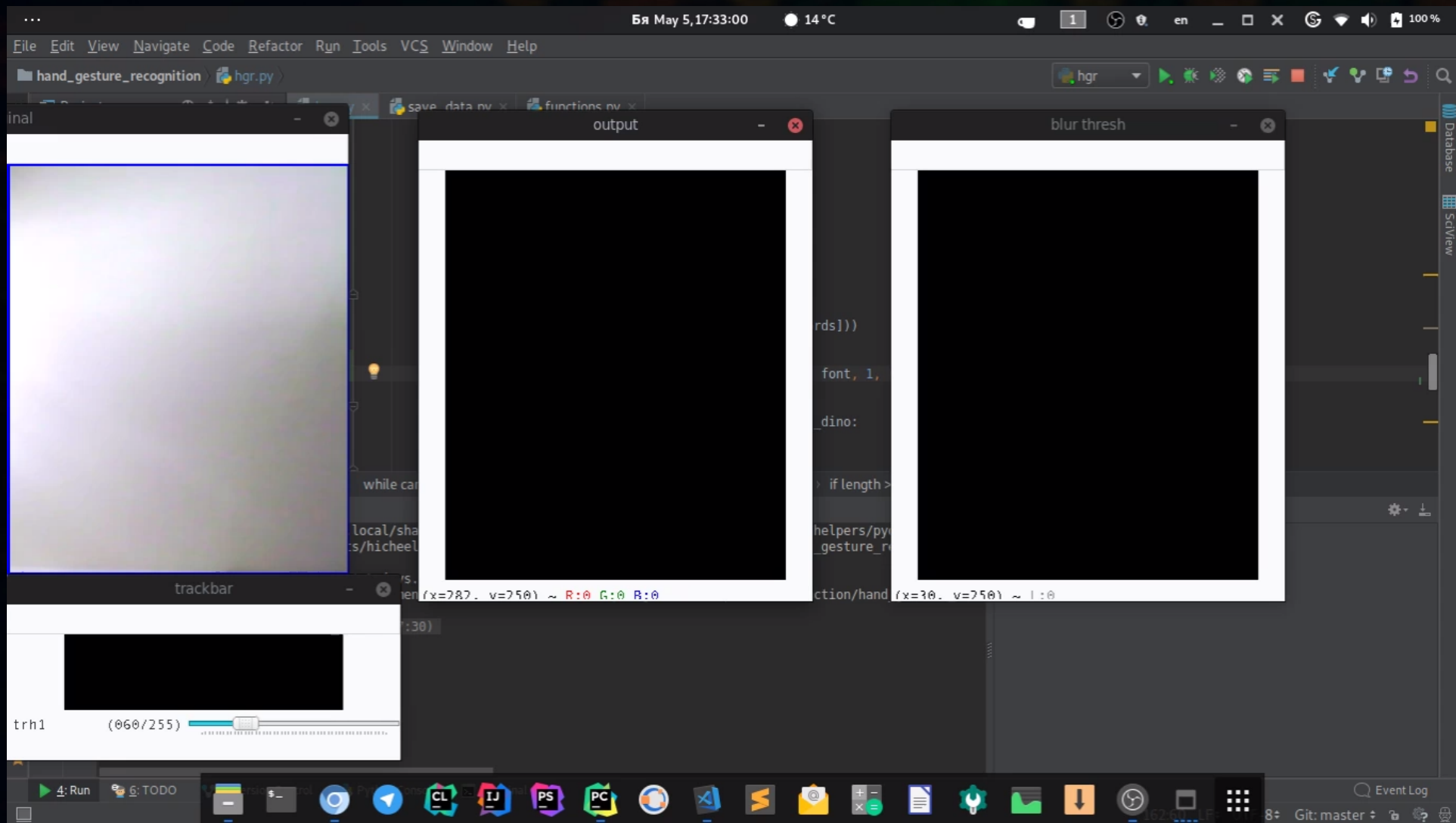
ERR\_INTERNET\_DISCONNECTED



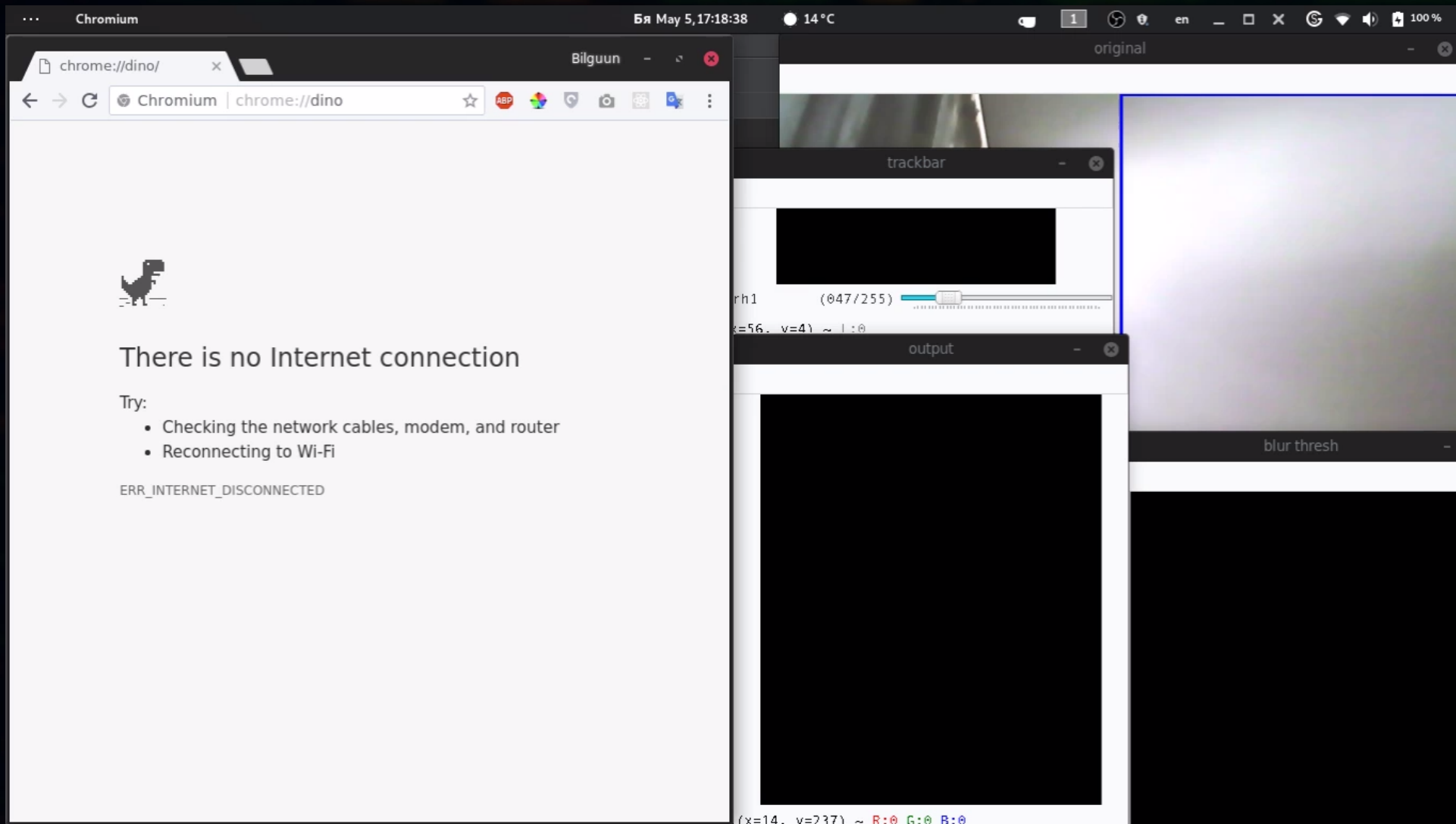
```
# PyAutoGUI
import pyautogui
# Queue
last_predict = deque([0, 0, 0])

# Play Dino
last_predict.popleft()
last_predict.append(data.clf.predict([coords]))
pred = func.most_common(last_predict)
if prev != pred:
    if prev == 0 and pred == 5 and status_dino:
        pyautogui.press('space')
    prev = pred
```

# ТУРШИЛТ



# ТУРШИЛТ (CHROME DINO)



# ҮР ДҮН

Хуруу	Танилтын хувь
0	95%
1	90%
2	70%
3	25%
4	90%
5	100%

# ДҮГНЭЛТ

Орчин болон гар хоорондоо ялгаатай өнгөтэй үед арын фоныг орж ирж буй зургаас хасахад гарыг бүрэн ялгаж авч чадсан. Threshold хийсэн зургаас findContours, convexHull, convexityDefects функцуудыг ашиглан гарын онцгой цэгүүдийг тооцоолж олсноор тэдгээрийг машин сургалтын SVM аргаар ангилан орж ирсэн зургаас гар ямар хэлбэр гаргаж байгааг олох боломжтой бөгөөд боломжийн үр дүнд хүрсэн. Цааш Convex hull аргаар олсон цэгүүдийг оновчтой хуваарилж, SVM-ын параметруудийг өөрчилж, тохиромжтой утгуудыг олсноор танилтыг сайжруулах боломжтой.

**АНХААРАЛ ТАВЬСАНД  
БАЯРЛАЛАА**