


CITS5503 Lab9

Wenxiao Zhang 22792191

[Step 1] Detecting Languages from text

[Step 1.1] Modify code

install `iso-639`

```
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$ pip3 install iso-639
Defaulting to user installation because normal site-packages is not writeable
Collecting iso-639
  Downloading iso-639-0.4.5.tar.gz (167 kB)
     167.4/167.4 kB 7.9 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: iso-639
  Building wheel for iso-639 (setup.py) ... done
  Created wheel for iso-639: filename=iso_639-0.4.5-py3-none-any.whl size=169061 sha256=4df21eecb7cf7bcfa0d7dd91d15
  a
  Stored in directory: /home/moebuta/.cache/pip/wheels/ed/ce/cc/1961a4de7090b2e92895fb087abfa0080a542a5706c5948bcc
Successfully built iso-639
Installing collected packages: iso-639
Successfully installed iso-639-0.4.5
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$
```

Write a script to recognize different languages and return the message in designated format.

```
comprehend.py X
2022s2 > cits5503 > labs > lab9 > comprehend.py > main
1 import boto3
2 import argparse
3 from iso639 import languages
4
5 client = boto3.client('comprehend')
6
7
8 def parse_args():
9     parser = argparse.ArgumentParser(description="arg parser")
10    parser.add_argument("-i", "--text", default=None, type=str)
11    return parser.parse_args()
12
13 def main():
14     args = parse_args()
15     if not args.text:
16         return
17
18     # Detect Entities
19     response = client.detect_dominant_language(
20         Text=args.text
21     )
22
23     language = languages.get(alpha2=response['Languages'][0]['LanguageCode']).name
24
25     confidence = str(int(response['Languages'][0]['Score'] * 100))
26
27     print(language+" detected with "+confidence+"%"+ " confidence")
28
29 if __name__ == "__main__":
30     main()
```

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE

```
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$ python3 comprehend.py -i "The French Revolution was a period of social and political upheaval in France and its colonies beginning in 1789 and ending in 1799"
English detected with 99% confidence
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$
```

[Step 1.2] Test code with other languages

Spanish:

```
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$ python3 comprehend.py -i "El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su primera parte con el título de El ingenioso hidalgo don Quijote de la Mancha a comienzos de 1605, es una de las obras más destacadas de la literatura española y la literatura universal, y una de las más traducidas. En 1615 aparecería la segunda parte del Quijote de Cervantes con el título de El ingenioso caballero don Quijote de la Mancha."
Spanish detected with 99% confidence
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$
```

French:

```
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$ python3 comprehend.py -i "Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits Je l'aime à mourir Vous pouvez détruire Tout ce qu'il vous plaira Elle n'a qu'à ouvrir l'espace de ses bras Pour tout reconstruire Pour tout reconstruire Je l'aime à mourir"
French detected with 99% confidence
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$
```

Italian:

```
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$ python3 comprehend.py -i "L'amor che move il sole e l'altre stelle."
Italian detected with 99% confidence
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$
```

[Step 2] Sentiment Analysis

Use boto3 and AWS comprehend to create a python script for sentiment analysis.

```
sentiment.py X
2022s2 > cits5503 > labs > lab9 > sentiment.py > ...
1  import boto3
2  from iso639 import languages
3
4  client = boto3.client('comprehend')
5
6  english = "The French Revolution was a period of social and political upheaval in France and its colonies beginning
7  spanish = "El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su primera parte con el t
8  french = "Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits Je l'aime à mourir
9  italian = "L'amor che move il sole e l'altre stelle."
10
11
12  def sentiment_analysis(text):
13      # Detect Entities
14      lan_response = client.detect_dominant_language(
15          Text=text
16      )
17      code = lan_response['Languages'][0]['LanguageCode']
18      language = languages.get(alpha2=code).name
19      sen_reponse = client.detect_sentiment(Text=text, LanguageCode = code)
20      sentiment = sen_reponse['Sentiment']
21      print("The sentiment of the %s text is %s" % (language, sentiment))
22
23  sentiment_analysis(english)
24  sentiment_analysis(spanish)
25  sentiment_analysis(french)
26  sentiment_analysis(italian)
27
PROBLEMS  OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$ python3 sentiment.py
The sentiment of the English text is NEUTRAL
The sentiment of the Spanish text is NEUTRAL
The sentiment of the French text is NEGATIVE
The sentiment of the Italian text is POSITIVE
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$
```

[Step 3] Repeat steps from [Step 2] for detecting entities.

python script:

```

sentiment.py  entity.py  X  keyphrase.py
2022s2 > cits5503 > labs > lab9 > entity.py > ...
1  import boto3
2  from iso639 import languages
3
4  client = boto3.client('comprehend')
5
6  english = "The French Revolution was a period of social and political upheaval in
7  spanish = "El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Pu
8  french = "Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du somme
9  italian = "L'amor che move il sole e l'altre stelle."
10
11
12  def entity_detection(text):
13      lan_response = client.detect_dominant_language(
14          Text=text
15      )
16      code = lan_response['Languages'][0]['LanguageCode']
17      language = languages.get(alpha2=code).name
18      ent_reponse = client.detect_entities(Text=text, LanguageCode = code)
19      entities = ent_reponse['Entities']
20      if not entities:
21          print('There is no entity in the %s text'%language)
22          return
23      print("The entities in the %s text are:" % language)
24      for ent in entities:
25          print(ent['Text'] + ' is ' + ent['Type'])
26      print()
27
28  entity_detection(english)
29  entity_detection(spanish)
30  entity_detection(french)
31  entity_detection(italian)
32

```

output:

```
● moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$ python3 entity.py
The entities in the English text are:
French Revolution is EVENT
France is LOCATION
1789 is DATE
1799 is DATE

The entities in the Spanish text are:
El Quijote is TITLE
Miguel de Cervantes Saavedra is PERSON
primera parte is QUANTITY
El ingenioso hidalgo don Quijote de la Mancha is TITLE
1605 is DATE
una is QUANTITY
española is OTHER
una de las más traducidas is QUANTITY
1615 is DATE
segunda parte is QUANTITY
Quijote de Cervantes is TITLE
El ingenioso caballero don Quijote de la Mancha is TITLE

The entities in the French text are:
aujourd'hui is DATE
Tout ce qu'il is QUANTITY
tout is QUANTITY
tout is QUANTITY

There is no entity in the Italian text
○ moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$ █
```

[Step 4] Repeat steps from [Step 2] for detecting keyphrases.

python script:

```

2022s2 > cits5503 > labs > lab9 > keyphrase.py > keyphrase_detection
1  import boto3
2  from iso639 import languages
3
4  client = boto3.client('comprehend')
5
6  english = "The French Revolution was a period of social and political upheaval in France an
7  spanish = "El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su
8  french = "Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses
9  italian = "L'amor che move il sole e l'altre stelle."
10
11 def keyphrase_detection(text):
12     lan_response = client.detect_dominant_language(
13         Text=text
14     )
15     code = lan_response['Languages'][0]['LanguageCode']
16     language = languages.get(alpha2=code).name
17     ph_reponse = client.detect_key_phrases(Text=text, LanguageCode = code)
18     phs = ph_reponse['KeyPhrases']
19     if not phs:
20         print('There is no key phrase in the %s text'%language)
21         return
22     print("The key phrases in the %s text are:" % language)
23     for ph in phs:
24         print(ph['Text'])
25     print()
26
27 keyphrase_detection(english)
28 keyphrase_detection(spanish)
29 keyphrase_detection(french)
30 keyphrase_detection(italian)
31

```

output:

```
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$ python3 keyphrase.py
The key phrases in the English text are:
The French Revolution
a period
social and political upheaval
France
its colonies
1789
1799

The key phrases in the Spanish text are:
El Quijote
la obra
más conocida
Miguel de Cervantes Saavedra
su primera parte
el título
El ingenioso hidalgo don Quijote de la Mancha
comienzos
1605
las obras
más destacadas
la literatura española
la literatura universal
las más traducidas
la segunda parte
Quijote de Cervantes
el título
ingenioso caballero don Quijote de la Mancha

The key phrases in the French text are:
Moi
je
n'étais rien
aujourd'hui
Je suis le gardien Du sommeil de ses nuits
Je
l'
Vous
Tout ce
qu'
il
vous
Elle
L'espace de ses bras
tout
tout
Je
l'

The key phrases in the Italian text are:
L'amor
che
il sole
l'altre stelle
```

[Step 5] Repeat steps from [Step 2] for detecting syntax.

python script:

```
sentiment.py  entity.py  keyphrase.py  syntax.py X
2022s2 > cits5503 > labs > lab9 > syntax.py > syntax_detection
1  import boto3
2  from iso639 import languages
3
4  client = boto3.client('comprehend')
5
6  english = "The French Revolution was a period of social and political upheaval in France and its
7  spanish = "El Quijote es la obra más conocida de Miguel de Cervantes Saavedra. Publicada su prim
8  french = "Moi je n'étais rien Et voilà qu'aujourd'hui Je suis le gardien Du sommeil de ses nuits
9  italian = "L'amor che move il sole e l'altre stelle."
10
11 def syntax_detection(text):
12     lan_response = client.detect_dominant_language(
13         Text=text
14     )
15     code = lan_response['Languages'][0]['LanguageCode']
16     language = languages.get(alpha2=code).name
17     syn_reponse = client.detect_syntax(Text=text, LanguageCode = code)
18     syntax = syn_reponse['SyntaxTokens']
19     print("The syntax in the %s text are:" % language)
20     output = ''
21     for syn in syntax:
22         output += syn['Text'] + ' is ' + syn['PartOfSpeech']['Tag'] + '\t'
23     print(output)
24     print()
25
26 syntax_detection(english)
27 syntax_detection(spanish)
28 syntax_detection(french)
29 syntax_detection(italian)
30
```

output:

```
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$ python3 syntax.py
The syntax in the English text are:
The is DET      French is PROPON Revolution is PROPON      was is VERB      a is DET      period is NOUN      of is ADP      social is ADJ
and is CONJ     political is ADJ      upheaval is NOUN      in is ADP      France is PROPON and is CONJ      its is PRON      colo
nies is NOUN      beginning is VERB      in is ADP      1789 is NUM      and is CONJ      ending is VERB      in is ADP      1799 is NU
M      . is PUNCT

The syntax in the Spanish text are:
El is DET      Quijote is PROPON      es is VERB      la is DET      obra is NOUN      más is ADV      conocida is ADJ de is ADP
Miguel is PROPON de is ADP      Cervantes is PROPON      Saavedra is PROPON      . is PUNCT      Publicada is VERB      su is DET
primera is ADJ parte is NOUN      con is ADP      el is DET      título is NOUN      de is ADP      El is DET      ingenioso is ADJ hi
dalgo is NOUN don is PROPON      Quijote is PROPON      de is ADP      la is DET      Mancha is PROPON a is ADP      comienzos is PRO
PN      de is ADP      1605 is NUM      , is PUNCT      es is VERB      una is PRON      de is ADP      las is DET      obras is NOUN
más is ADV      destacadas is ADJ      de is ADP      la is DET      literatura is NOUN      española is ADJ y is CCONJ      la i
s DET      literatura is NOUN      universal is ADJ      , is PUNCT      y is CCONJ      una is PRON      de is ADP      las is DET
más is ADV      traducidas is ADJ      . is PUNCT      En is ADP      1615 is NUM      aparecería is VERB      la is DET
segunda is ADJ parte is NOUN      del is ADP      Quijote is PROPON      de is ADP      Cervantes is PROPON      con is ADP      el is
DET      título is NOUN      de is ADP      El is DET      ingenioso is ADJ      caballero is NOUN      don is PROPON      Quijote is P
ROPON      de is ADP      la is DET      Mancha is PROPON . is PUNCT

The syntax in the French text are:
Moi is PRON      je is PRON      n' is ADV      étais is AUX      rien is PRON      Et is CCONJ      voilà is VERB      qu' is SCONJ      aujourd
'hui is ADV      Je is PRON      suis is AUX      le is DET      gardien is NOUN Du is ADP      sommeil is NOUN de is ADP      ses
is DET      nuits is NOUN      Je is PRON      l' is PRON      aime is VERB      à is ADP      mourir is VERB      Vous is PRON      pouvez is
AUX      détruire is VERB      Tout is DET      ce is PRON      qu' is PRON      il is PRON      vous is PRON      plaira is VERB      Elle is
PRON      n' is ADV      a is VERB      qu' is ADV      à is ADP      ouvrir is VERB      L' is DET      espace is NOUN      de is ADP
ses is DET      bras is NOUN      Pour is ADP      tout is PRON      reconstruire is VERB      Pour is ADP      tout is PRON      reconstruire
is VERB      Je is PRON      l' is PRON      aime is VERB      à is ADP      mourir is VERB

The syntax in the Italian text are:
L' is DET      amor is NOUN      che is PRON      move is VERB      il is DET      sole is NOUN      e is CCONJ      l' is DET      altre i
s ADJ      stelle is NOUN      . is PUNCT
```


[Step 6] In an S3 bucket add some images to test algorithms.

Upload images to the bucket via AWS console

Files and folders (4 Total, 419.9 KB)

Remove

Add files

Add folder

All files and folders in this table will be uploaded.

Find by name

< 1 >

<input type="checkbox"/>	Name ▲	Folder ▼	Type ▼	Size ▼
<input type="checkbox"/>	Urban setting.jpg	-	image/jpeg	258.3 KB
<input type="checkbox"/>	a person on the beach.jpg	-	image/jpeg	85.2 KB
<input type="checkbox"/>	image with text.jpg	-	image/jpeg	26.9 KB
<input type="checkbox"/>	people showing their faces.jpg	-	image/jpeg	49.5 KB

Destination

Destination

s3://22792191-cloudstorage

► Destination details

Bucket settings that impact new objects stored in the specified destination.

► Permissions

Grant public access and access to other AWS accounts.

► Properties

Specify storage class, encryption settings, tags, and more.

Cancel

Upload

Summary

Destination

s3://22792191-cloudstorage

Succeeded

✔ 4 files, 419.9 KB (100.00%)

Failed

⌵ 0 files, 0 B (0%)

Files and folders

Configuration

Files and folders (4 Total, 419.9 KB)

🔍 Find by name

< 1 >

Name ▲	Folder ▼	Type ▼	Size ▼	Status ▼	Error
Urban setting.jpg	-	image/jpeg	258.3 KB	✔ Succeeded	-
a person on the beach.jpg	-	image/jpeg	85.2 KB	✔ Succeeded	-
image with text.jpg	-	image/jpeg	26.9 KB	✔ Succeeded	-
people showing their faces.jpg	-	image/jpeg	49.5 KB	✔ Succeeded	-

[Step 7] Create scripts using boto3 and rekognition to test label recognition, image moderation, facial analysis and extracting text from images.

Label recognition:

```
label.py  X  moderation.py  facial.py
2022s2 > cits5503 > labs > lab9 > label.py > ...
1  import boto3
2  client = boto3.client('rekognition')
3
4  imgs=[
5      {
6          'S3Object': {
7              'Bucket': '22792191-cloudstorage',
8              'Name': 'Urban setting.jpg'
9          },
10     },
11     {
12         'S3Object': {
13             'Bucket': '22792191-cloudstorage',
14             'Name': 'a person on the beach.jpg'
15         },
16     },
17     {
18         'S3Object': {
19             'Bucket': '22792191-cloudstorage',
20             'Name': 'image with text.jpg'
21         },
22     },
23     {
24         'S3Object': {
25             'Bucket': '22792191-cloudstorage',
26             'Name': 'people showing their faces.jpg'
27         },
28     },
29 ]
30
31
32 def label_detection(img):
33     response = client.detect_labels(Image=img, MaxLabels=3, MinConfidence=0.95)
34     print('label detection for %s:' % img['S3Object']['Name'])
35     print(response['Labels'])
36     print()
37
38 label_detection(imgs[0])
39 label_detection(imgs[1])
40 label_detection(imgs[2])
41 label_detection(imgs[3])
42
```

output:

```
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$ python3 label.py
label detection for Urban setting.jpg:
[{'Name': 'Grass', 'Confidence': 99.95657348632812, 'Instances': [], 'Parents': []}, {'Name': 'City', 'Confidence': 98.78321838378906, 'Instances': [], 'Parents': [{'Name': 'Urban'}]}, {'Name': 'Urban', 'Confidence': 98.78321838378906, 'Instances': [], 'Parents': []}]

label detection for a person on the beach.jpg:
[{'Name': 'Shorts', 'Confidence': 99.38862609863281, 'Instances': [], 'Parents': []}, {'Name': 'Person', 'Confidence': 98.74070739746094, 'Instances': [{'BoundingBox': {'Width': 0.2898404002189636, 'Height': 0.7046762108802795, 'Left': 0.46094265580177307, 'Top': 0.17930786311626434}, 'Confidence': 98.74070739746094}], 'Parents': []}, {'Name': 'Water', 'Confidence': 91.67666625976562, 'Instances': [], 'Parents': []}]

label detection for image with text.jpg:
[{'Name': 'Water', 'Confidence': 97.87112426757812, 'Instances': [], 'Parents': []}, {'Name': 'Sea', 'Confidence': 96.966552734375, 'Instances': [], 'Parents': [{'Name': 'Water'}]}, {'Name': 'Shoreline', 'Confidence': 96.96572875976562, 'Instances': [], 'Parents': [{'Name': 'Water'}]}]

label detection for people showing their faces.jpg:
[{'Name': 'Person', 'Confidence': 99.44589233398438, 'Instances': [{'BoundingBox': {'Width': 0.4016018807888031, 'Height': 0.7247241735458374, 'Left': 0.144425630569458, 'Top': 0.2644144892692566}, 'Confidence': 99.44589233398438}, {'BoundingBox': {'Width': 0.5290720462799072, 'Height': 0.8590338826179504, 'Left': 0.46075254678726196, 'Top': 0.11579621583223343}, 'Confidence': 99.05853271484375}], 'Parents': []}, {'Name': 'Clothing', 'Confidence': 95.78443145751953, 'Instances': [], 'Parents': []}, {'Name': 'Sleeve', 'Confidence': 93.77432250976562, 'Instances': [], 'Parents': [{'Name': 'Clothing'}]}]
```

Image moderation:

```
extraction.py  moderation.py X
2022s2 > cits5503 > labs > lab9 > moderation.py > ...
1  import boto3
2  from chardet import detect
3  client = boto3.client('rekognition')
4
5  imgs=[
6      {
7          'S3Object': {
8              'Bucket': '22792191-cloudstorage',
9              'Name': 'Urban setting.jpg'
10         },
11     },
12     {
13         'S3Object': {
14             'Bucket': '22792191-cloudstorage',
15             'Name': 'a person on the beach.jpg'
16         },
17     },
18     {
19         'S3Object': {
20             'Bucket': '22792191-cloudstorage',
21             'Name': 'image with text.jpg'
22         },
23     },
24     {
25         'S3Object': {
26             'Bucket': '22792191-cloudstorage',
27             'Name': 'people showing their faces.jpg'
28         },
29     },
30 ]
31
32
33 def img_moderation(img):
34     response = client.detect_moderation_labels(Image=img)
35     print('image moderation for %s:' % img['S3Object']['Name'])
36     print(response['ModerationLabels'])
37     print()
38
39 img_moderation(imgs[0])
40 img_moderation(imgs[1])
41 img_moderation(imgs[2])
42 img_moderation(imgs[3])
```

output:

```
PROBLEMS  OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE
● moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$ python3 moderation.py
image moderation for Urban setting.jpg:
[]

image moderation for a person on the beach.jpg:
[{'Confidence': 86.2708969116211, 'Name': 'Suggestive', 'ParentName': ''}, {'Confidence': 86.2708969116211, 'Name': 'Barecheste
d Male', 'ParentName': 'Suggestive'}]

image moderation for image with text.jpg:
[]

image moderation for people showing their faces.jpg:
[]
```

Facial analysis:

```
label.py  moderation.py  facial.py  X
2022s2 > cits5503 > labs > lab9 > facial.py > ...
1  import boto3
2  client = boto3.client('rekognition')
3
4  imgs=[
5      {
6          'S3Object': {
7              'Bucket': '22792191-cloudstorage',
8              'Name': 'Urban setting.jpg'
9          },
10     },
11     {
12         'S3Object': {
13             'Bucket': '22792191-cloudstorage',
14             'Name': 'a person on the beach.jpg'
15         },
16     },
17     {
18         'S3Object': {
19             'Bucket': '22792191-cloudstorage',
20             'Name': 'image with text.jpg'
21         },
22     },
23     {
24         'S3Object': {
25             'Bucket': '22792191-cloudstorage',
26             'Name': 'people showing their faces.jpg'
27         },
28     },
29 ]
30
31
32 def facial_analysis(img):
33     response = client.detect_faces(Image=img)
34     print('facial analysis for %s:' % img['S3Object']['Name'])
35     print(response['FaceDetails'])
36     print()
37
38 facial_analysis(imgs[0])
39 facial_analysis(imgs[1])
40 facial_analysis(imgs[2])
41 facial_analysis(imgs[3])
42
```

output:

```
moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$ python3 facial.py
facial analysis for Urban setting.jpg:
[]

facial analysis for a person on the beach.jpg:
[{'BoundingBox': {'Width': 0.04054493084549904, 'Height': 0.07278057932853699, 'Left': 0.5491769909858704, 'Top': 0.20082081854343414}, 'Landmarks': [{'Type': 'eyeLeft', 'X': 0.5622652173042297, 'Y': 0.22048966586589813}, {'Type': 'eyeRight', 'X': 0.5760547518730164, 'Y': 0.228282168507576}, {'Type': 'mouthLeft', 'X': 0.5556396842002869, 'Y': 0.24419769644737244}, {'Type': 'mouthRight', 'X': 0.5670392513275146, 'Y': 0.25070884823799133}, {'Type': 'nose', 'X': 0.5631179809570312, 'Y': 0.2294585108757019}], 'Pose': {'Roll': 18.016693115234375, 'Yaw': -21.995115280151367, 'Pitch': 24.85446548461914}, 'Quality': {'Brightness': 86.16284942626953, 'Sharpness': 12.848764419555664}, 'Confidence': 99.86874389648438}]

facial analysis for image with text.jpg:
[]

facial analysis for people showing their faces.jpg:
[{'BoundingBox': {'Width': 0.1466192603111267, 'Height': 0.27660509943962097, 'Left': 0.49111127853393555, 'Top': 0.1619311422109604}, 'Landmarks': [{'Type': 'eyeLeft', 'X': 0.5263110995292664, 'Y': 0.30495426058769226}, {'Type': 'eyeRight', 'X': 0.5802080631256104, 'Y': 0.27094435691833496}, {'Type': 'mouthLeft', 'X': 0.559465229511261, 'Y': 0.3857216238975525}, {'Type': 'mouthRight', 'X': 0.6047527194023132, 'Y': 0.35760706663131714}, {'Type': 'nose', 'X': 0.5633824467658997, 'Y': 0.34961020946502686}], 'Pose': {'Roll': -26.257678985595703, 'Yaw': -12.550137519836426, 'Pitch': -7.0442376136779785}, 'Quality': {'Brightness': 97.44495391845703, 'Sharpness': 73.32209777832031}, 'Confidence': 99.99781799316406}, {'BoundingBox': {'Width': 0.1224934384226799, 'Height': 0.26064708828926086, 'Left': 0.31655120849609375, 'Top': 0.3192150592803955}, 'Landmarks': [{'Type': 'eyeLeft', 'X': 0.3420851230621338, 'Y': 0.43106698989868164}, {'Type': 'eyeRight', 'X': 0.3957558870315552, 'Y': 0.42059969902038574}, {'Type': 'mouthLeft', 'X': 0.35612332820892334, 'Y': 0.5134368538856506}, {'Type': 'mouthRight', 'X': 0.40099501609802246, 'Y': 0.5050134062767029}, {'Type': 'nose', 'X': 0.36642101407051086, 'Y': 0.47437113523483276}], 'Pose': {'Roll': -8.853300094604492, 'Yaw': -10.199845314025879, 'Pitch': -0.08856722712516785}, 'Quality': {'Brightness': 95.06905364990234, 'Sharpness': 67.22731018066406}, 'Confidence': 99.9968032836914}]

moebuta@Lenovo-MoeBuTa:~/2022s2/cits5503/labs/lab9$
```


Extracting text from images:

extraction.py X

2022s2 > cits5503 > labs > lab9 > extraction.py > ...

```
1  import boto3
2  client = boto3.client('rekognition')
3
4  imgs=[
5      {
6          'S3Object': {
7              'Bucket': '22792191-cloudstorage',
8              'Name': 'Urban setting.jpg'
9          },
10     },
11     {
12         'S3Object': {
13             'Bucket': '22792191-cloudstorage',
14             'Name': 'a person on the beach.jpg'
15         },
16     },
17     {
18         'S3Object': {
19             'Bucket': '22792191-cloudstorage',
20             'Name': 'image with text.jpg'
21         },
22     },
23     {
24         'S3Object': {
25             'Bucket': '22792191-cloudstorage',
26             'Name': 'people showing their faces.jpg'
27         },
28     },
29 ]
30
31
32 def text_extraction(img):
33     response = client.detect_text(Image=img)
34     print('text extraction for %s:' % img['S3Object']['Name'])
35     print(response['TextDetections'])
36     print()
37
38 text_extraction(imgs[0])
39 text_extraction(imgs[1])
40 text_extraction(imgs[2])
41 text_extraction(imgs[3])
```

