

# Euro Currency Note Identification Using AWS Sage Maker

# **Cloud Computing SS2022**

Submitted by:

Moeez Ur Rehman Muhammad Hasseb Anwar Sharish Kanwal Harmain Haider

Under the guidance of:

Prof. Dr. Christian Baun

# **Table of Content**

1.	INTRODU	UCTION	3
2.	Technolog	gies	4
	2.1	AWS SageMaker	4
	2.2	SageMaker End Point	4
	2.3	AWS lambda Functions	4
	2.4	AWS API Gatway	4
	2.5	AWS Amplify	4
	2.6	Flow Diagram	5
3.	Training I	Model On SageMaker	6
	3.1.1	Creating S3 bucket	6
	3.1.2	Uploading training images on S3	8
	3.1.3	Creating Notebook Instance on Sagemaker	9
	3.1.4	Image Classification on Sagemaker	13
	3.1.5	Deploying endpoint on Sagemaker	21
	3.1.	Testing the model from notebook	22
	3.1.7	Cleanup Sagemaker	24
4.	Web Appl	lication	25
5.	Predicting	g Currency note from webapp	28

## 1. Introduction

The goal of the project is to implement machine learning model on the cloud service, we have used AWS cloud platform for this project, The problem statement was to identify different currency notes, our training data is stored on AWS S3 from where is being pulled by AWS's Sagemaker service to train our model.

After the training the endpoint is being deployed on sagemaker which then be used by our webapp to predict out currency notes. For the sake of simplicity, we have trained our model for only Euro 5,10,20 and 50 notes.

To train our model we have used **Sagemaker's built in Image classification Algorithm** which is based on **Supervised Learning** that supports multi-class classification. It uses Conventional Neural Networks (CNN). More info on the image classification model of sagemaker can be access using this link <a href="https://docs.AWS.amazon.com/sagemaker/latest/dg/image-classification.html">https://docs.AWS.amazon.com/sagemaker/latest/dg/image-classification.html</a>



# 2. Technologies

# 2.1 AWS SageMaker

It is fully managed machine learning services. Machine learning models easily train, build and deployed directly into a production ready hosted environment with Sage Maker. It provides Jupyter notebook instance for easy access to your data sources for exploration and analysis, so you don't have to manage servers. It also provides common machine learning algorithms that we can run efficiently against extremely large data in a distributed environment. With native support we can also bring-our-own-algorithms and frameworks, SageMaker offers flexible distributed training options that adjust to your specific systems. Deploy a model into a secure and scalable environment by launching it with a few clicks from SageMaker Studio or the SageMaker console. Training and hosting are billed by minutes of usage, with no minimum fees and no upfront commitments.

# 2.2 SageMaker End point

An Amazon SageMaker endpoint is a fully managed service that allows you to make real-time inferences via a REST API. Taking the pain away from running your own EC2 instances, loading artefacts from S3, wrapping the model in some lightweight REST application, attaching GPUs and much more. This is great as it means with a single click or command you have a fully working solution.

# 2.3 AWS Lambda functions

When we write our code, we are responsible for our code only. Lambda manages network, memory, CPU and other resources to run code. Lambda manages resources because we cannot log in to the compute instances or customize the operating system on provided runtimes. On our behave lambda perform administrative and operational activities such as monitoring, managing capacity etc. By using Lambda API we invoke our lambda function.

We can use Lambda to:

Create our own backend that operates at AWS scales, performance and security. Build data processing triggers for AWS services such as Amazon Simple Storage Services (Amazon S3)

# 2.4 AWS API Gateway

It is a service for creating, maintaining, publishing and securing REST, HTTP, and WebSocket APIs at any scale. API developers can create APIs that access AWS or other web services, as well as data stored in the AWS Cloud. As an API Gateway API developer, you can create APIs for use in your own client applications.

### **API endpoint**

API endpoints are the specific digital location to retrieve the digital resource that exists there when request for information are sent by one program. To guarantee the proper functioning of the incorporated software, Endpoint specify where APIs can access resources.

# 2.5 AWS Amplify

AWS Amplify include ready to use components, code lines and built-in command line interface designed to help developers easily create and launch apps. It also allows you to securely and quickly integrate a wild range of functions ranging from API to AI. It is a full stack application platform with both client side and server-side code.

Various AWS services used in our project, to train the model we have used S3 to store our data and Sagemaker to train our model

### 2.6 Flow Diagram



User

### 3. Training model on Sagemake

aws iii Services Q Search for services, features, blogs, docs, and more	[Alt+S]	♦ 🛛 Stockholm ▼	MuhammadHaseebAnwar 🔻
Console Home Info		Actions <b>v</b>	Ũ
Recently visited Info		Welcome to AWS :	
<ul> <li>VPC</li> <li>API Gateway</li> <li>Lambda</li> <li>Amazon SageMaker</li> <li>CloudWatch</li> <li>AWS Cost Explorer</li> <li>S3</li> </ul>	IAM       Image: Resource Groups & Tag Editor       Image: Amazon Augmented Al       Image: EC2	Getting started with AWS [2] Learn the fundamentals and find valuable information to get the most out of AWS. Training and certification [2] Learn from AWS experts and advance your skills and knowledge. What's new with AWS? [2] Discover new AWS services, features, and	
View a	all services	P Regions.	
AWS Health Info :     Open issues	Cost and usage Info	:	

# 3.1.1 Creating S3 bucket

The very first task is to create a S3 bucket, we have to provide a name of the bucket, choose the region, in our case we have chosen eu-central-1 location which is in Frankfurt.

	s3.console.aws.amazon.com/s3/bucket/create?region=eu-central-1		E 🕯 🚺 🧯	) 🚥 🖉 🗯 🖬 🕼 🤫 :
Serv	ces Q Search for services, features, blogs, docs, and more	[Alt+5]	🗘 🕜 Global 🔻	MuhammadHaseebAnwar 🔻
Amazon	S3 > Buckets > Create bucket			٤
Crea Buckets	te bucket Info are containers for data stored in S3. Learn more 🔀			
Gen	eral configuration			
Bucke	it name			
clou	d-computing-dataset			
Bucke	name must be unique and must not contain spaces or uppercase letters. See rules for bucket	naming 🖸		
AWS	Region			
	Frankfurt) av santral 1			

Here we need to implement the object ownership, if we want the object of this bucket to be owned by multiple accounts then we should choose ACLs Enabled, Although the recommended property is **ACLs disabled**.

#### Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)
 All objects in this bucket are owned by this account.
 Access to this bucket and its objects is specified using only policies.

ACLs enabled

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

# Object Ownership

Bucket owner enforced

Here we can implement the security of our bucket, it is always recommended to block all public access of the bucket.

Public ensure and its applica custom	access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your trions will work correctly without public access. If you require some level of public access to this bucket or objects within, you can ize the individual settings below to suit your specific storage use cases. Learn more
J Bl	<b>ock <i>all</i> public access</b> rning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
- 0	Block public access to buckets and objects granted through <i>new</i> access control lists (ACLs) S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- 🗸	Block public access to buckets and objects granted through any access control lists (ACLs) S3 will ignore all ACLs that grant public access to buckets and objects.
- 🗸	Block public access to buckets and objects granted through <i>new</i> public bucket or access point policies S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
_ ~	Block public and cross-account access to buckets and objects through <i>any</i> public bucket or access point policies S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects

#### Just click on create bucket and our bucket will be created.

Advanced settings		
After creating the bucket you can upload files and folders to the	he bucket, and configure additiona	l bucket settings.
	Cancel	Create bucket

# 3.1.2 Uploading training images on s3

Next step is to create folders within the S3 bucket. We have created our main folder with the name 'CC\_Dataset'

mazon S3 > Buckets > cloud-co	omputing-dataset								
loud-computing-d	lataset Info								
Objects Properties Pe	rmissions Met	trics Manageme	ent Access P	oints					
Objects (1)									
Objects (1) Objects are the fundamental entities sto grant them permissions. Learn more	red in Amazon S3. You o	can use Amazon S3 inver	ntory 🔀 to get a list	of all objects in yo	ur bucket. For others	to access y	our objects, you	u'll need to expli	citly
Objects (1) Objects are the fundamental entities sto grant them permissions. Learn more 🖸	red in Amazon S3. You o	can use Amazon S3 inver	ntory 🛃 to get a list	of all objects in yo Delete	ur bucket. For others	to access y	our objects, you <b>ite folder</b>	u'll need to expli	citly
Objects (1) Objects are the fundamental entities sto grant them permissions. Learn more C C C Copy S3 URI	red in Amazon S3. You o	can use Amazon S3 inver	ntory 🕻 to get a list	of all objects in yo Delete	Actions	to access y	our objects, you I <b>te folder</b>	u'll need to expli	citly d
Objects (1) Objects are the fundamental entities sto grant them permissions. Learn more C C C C C C C C C C C C C C C C C C C	red in Amazon S3. You o	can use Amazon S3 inver	open	of all objects in yo Delete	ur bucket. For others	to access y	our objects, you I <b>te folder</b>	u'll need to expli	citly d
Objects (1) Objects are the fundamental entities sto grant them permissions. Learn more C C C C C C C C C C C C C C C C C C C	red in Amazon S3. You o	can use Amazon S3 inver	open	of all objects in yo Delete	Actions V	to access y Crea	our objects, you ite folder Storage cla	u'll need to expli	citly nd ③
Objects (1) Objects are the fundamental entities sto grant them permissions. Learn more  C C C C C C C C C C C C C C C C C C C	red in Amazon S3. You o	can use Amazon S3 inver	open Z	of all objects in yo Delete	Actions V Size	to access y Crea	our objects, you ite folder Storage cla	u'll need to expli	d Ø Ø Ø

Now, we need to gather our training data, for this model we have taken around 150 sample images of each 5,10,20 and 50 Euro currency notes.

Create folder for each label as shown in picture below.

Amazon S3	> Buckets > cloud-comp	outing-dataset > CC-Datas	et/			
CC-Da	ataset/				٥	Copy S3 URI
Objects	Properties					
Object	<b>ts</b> (8)					
Objects a them peri	re the fundamental entities stored i missions. Learn more 🔀	n Amazon S3. You can use <b>Amazo</b>	n S3 inventory 🔀 to get a list	t of all objects in your bucket. For others	to access your objects, you'll need to expl	icitly grant
C	්ට Copy S3 URI ර්ට	Copy URL 🕑 Dowr	iload Open 🖸	Delete Actions <b>v</b>	Create folder IPlo	ad
Q Fin	ad objects by prefix				< 1	> ©
	Name	▲ Туре		⊽ Size		$\nabla$
	EURO-10/	Folder	1.51		<b>1</b> 72 ( <b>1</b> 5	
	EURO-20/	Folder	-			
	EURO-5/	Folder	~			
	EURO-50/	Folder				

Now upload sample images in respective folders.

aws 🔛 Services Q Search for se	ervices, features, blogs, docs, and more	[Alt+S]		🗘 🕐 Global	▼ MuhammadHaseebAnwar ▼
Amazon S3 ×	Amazon S3 > Buckets > cloud-com	puting-dataset > CC-Dataset/ >	EURO-10/		
Buckete	EURO-10/				Copy S3 URI
Access Deinte					
Access Points	Objects Properties				
Multi Dagion Access Points					
Patch Operations					
batch Operations	Objects (144)				
Access analyzer for 55	Objects are the fundamental entities stored	in Amazon S3. You can use Amazon S3 inven	tory 🔀 to get a list of all objects in your bucket. For other	s to access your objects, you	'll need to explicitly grant
	them permissions. Learn more				
Block Public Access settings for	C C Copy S3 URI	Copy URL 🕑 Download	Open 🖾 Delete Actions 🔻	Create folder	🖪 Upload
this account	Eind objects his prefix				< 1 > @
Storage Lens	a Pillo objects by prenk				
Dashboards	Name	▲ Type マ	Last modified	⊽ Size ⊽	Storage class  □
AWS Organizations settings	IMG_20220618_140650.j	ipg jpg	June 18, 2022, 19:01:47 (UTC+02:00)	8.9 KB	Standard
	□ MG_20220618_140652.j	ipg jpg	June 18, 2022, 19:01:47 (UTC+02:00)	8.5 KB	Standard
Feature spotlight 3	IMG_20220618_140653.j	ipg jpg	June 18, 2022, 19:01:47 (UTC+02:00)	8.9 KB	Standard
	IMG_20220618_140654.j	ipg jpg	June 18, 2022, 19:02:22 (UTC+02:00)	9.6 KB	Standard
AWC Masketalace for 57	IMG_20220618_140655.j	jpg jpg	June 18, 2022, 19:02:22 (UTC+02:00)	9.6 KB	Standard
· And harkequate for 55	IMG_20220618_140656.j	ipg ipg	June 18, 2022, 19:02:23 (UTC+02:00)	8.0 KB	Standard
	IMG_20220618_140657.j	ipg jpg	June 18, 2022, 19:02:23 (UTC+02:00)	8.3 KB	Standard
	IMG_20220618_140659.j	ipg ipg	June 18, 2022, 19:02:23 (UTC+02:00)	8.3 KB	Standard

# 3.1.3 Creating notebook instance on sagemker

Now as we have setup our S3 bucket and folders as we need to train our model, now we are moving towards our AWS service called Sagemaker.

This is the landing page of our Sagemaker.

aws Services Q Search for services, feature	ures, blogs, docs, and more [Alt+S]	🗘 🕜 Stockholm 🔻 MuhammadHaseebAnwar 🔻 🕯
Amazon SageMaker $ imes$	MACHINE LEARNING	
Getting started	Amazon SageMaker	New to SageMaker?
Control panel Studio	Build, train, and deploy	Get started with Amazon SageMaker by completing the quick start guide.
Studio Lab 🔁 🛛 NEW RStudio 🛛 NEW	machine learning models at	Get Started
SageMaker dashboard	SCALE The aukkest and essiest way to aet ML models from idea to production.	
Images Search		Documentation
Notebook     Processing	How it works	Getting started 🔀
► Training	What is Amazon SageMaker?	Tutorials Documentation 🖸
Inference     AWS Marketplace	Amazon SageMaker provides machine learning (ML) capabilities for data scientists and developers to prepare, build, train, and deploy high-quality ML models efficiently.	Developer Resources 🔀
P And Parketplace	New user onboarding guide (New)	AWS Developer Forum
Take survey 🗹 🗸 🗸	Get started with Amazon SageMaker by completing the quick start	contact us E

### We need to create Notebook instance within sagemaker, choose your desired region, we have chosen eu-central-

Studio Lab EZ NEW	Amazon SageMaker > N	lotebook instances		US East (N. Virginia)	us-east-1	i i		
RStudio NEW	Notebook instanc	PS		US East (Ohio)	us-east-2	tebool	c instance	
	O Search notebook in	stances		US West (N. California)	us-west-1	< 1		0
ageMaker dashboard	C Statel Hotesook in			US West (Oregon)	us-west-2			~
Images	Name		Creation time			ions		
Search			There are currently no resources	Africa (Cape Town)	af-south-1			
Notebook			mare the contentity no resources	Acia Dacific (Hong Kong)	an eact 1	-		
Notebook instances				Asia Facilic (Holig Kolig)	ap-east- i			
Lifecycle configurations				Asia Pacific (Jakarta) a	p-southeast-3			
Git repositories				Asia Pacific (Mumbai)	ap-south-1			
Processing				Asia Pacific (Osaka) a	p-northeast-3			
Training				Asia Pacific (Seoul) a	p-northeast-2			
Inference				Asia Pacific (Singapore) a	p-southeast-1	4		
				Asia Pacific (Svdnev) a	p-southeast-2	I		
AWS Marketplace				Asia Dasifia (Talasa)		I		
				Азіа Расіпс (Токуо) а	ip-northeast- i	I		
ke survey 🔀				Canada (Central)	ca-central-1	I		
torials						I		
ocumentation 🗠				Europe (Frankfurt)	eu-central-1			

services, features, blogs, docs, and more	[Alt+S]	D	¢	0	Frankfurt 🔻	Muhamma	adHaseebAn	war 🔻
Amazon SageMaker > Notebook instances								
Notebook instances		C	Actio	ons 🔻	Create n	otebook ins	tance	
Q Search notebook instances						< 1	0	

Give name of your notebook instance, and choose your instance type. Further information on sage maker instance types is provided here <u>https://AWS.amazon.com/sagemaker/pricing/</u>.

If you are using Sagemaker for the first time, then take advantage of the free tier, Free tier information is also provided in the above link.

aws	Services	<b>Q</b> Search for services, features, blogs, docs, and more	[Alt+S]				
=	Amazon Sage	Maker > Notebook instances > Create notebook instance					
	Create	notebook instance					
	Amazon SageMaker provides pre-built fully managed notebook instances that run Jupyter notebooks. The include example code for common model training and hosting exercises. Learn more 🔀						
	Noteboo	ok instance settings					
	Notebook	instance name					
	Cloud-Co	mputing-Project-Image-Recognition-NB-Instance					
	Maximum of	63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be un	ique within your account in an AWS Region.				
	Notebook	instance type					
	ml.t3.me	dium	•				
	Platform io	dentifier Learn more 🖸					
	Amazon	Linux 2, Jupyter Lab 1	•				
	► Additi	onal configuration					

In this step we can create a new IAM role or use already created IAM role. IAM role is nothing but a set of permissions, as you need to access to different other services of AWS within this notebook, make sure that the role which you select here has the necessary permissions to access those web services. Further information of IAM roles are provided here <a href="https://docs.AWS.amazon.com/IAM/latest/UserGuide/id\_roles.html">https://docs.AWS.amazon.com/IAM/latest/UserGuide/id\_roles.html</a>

Root access: you really don't need to give root access to the sagemaker notebook here if you are working on production environment, root access means administrative privileges, which means by using this notebook you can edit, remove any files on the system.

IAM role Notebook instances require permissions to call other services including SageMak AmazonSageMakerFullAccess IAM policy attached.	er and S3. Choose a role or let us create a role with the
AmazonSageMakerServiceCatalogProductsUseRole	
Root access - optional	
Enable - Give users root access to the notebook	
<ul> <li>Disable - Don't give users root access to the notebook</li> <li>Lifecycle configurations always have root access</li> </ul>	
Encryption key - optional Encrypt your notebook data. Choose an existing KMS key or enter a key's ARN.	
No Custom Encryption	•
▶ Network - optional	
<ul> <li>Network - optional</li> <li>Git repositories - optional</li> </ul>	
<ul> <li>Network - optional</li> <li>Git repositories - optional</li> <li>Tags - optional</li> </ul>	

After our notebook instance is created, we need to start this notebook instance after that we are going to write our model.

Am2700							- Idildi	
Not	ebook instances			C	Actions 🔻	Create n	otebook	instance
Q	Search notebook instances						< 1	> @
	Name	▽	Instance	Creation time		Status	$\bigtriangledown$	Actions
0	Cloud-Computing-Project-Image-Recognition-NB-Instance		ml.t2.medium	May 29, 2022 13:3	5 UTC	⊖ Stoppe	Ы	Start

ices, features, blogs, docs, and more	[Alt+S]			۶.	¢	0	Frankfurt 🔻	MuhammadHaseebAnw
Amazon SageMaker > Notebook instances								
Notebook instances				C	Actio	ns 🔺	Create no	tebook instance
Q Search notebook instances					Open Open	Jupyter JupyterLa	b	< 1 > ③
Name		▽	Instance	Creation time	Stop		atus	
Cloud-Computing-Project-Image-	Recognition-NB-Instance		ml.t2.medium	May 29, 2022	1 Upda	te settings	Stopped	Start
					Add/ Delet	Edit tags e		

# 3.1.4 Image classification on Sagemaker

### Sagemaker Model:

We have used **Sagemaker's built in Image classification Algorithm** which is based on **Supervised Learning** that supports multi-class classification. It uses Conventional Neural Networks (CNN). More info on the image classification model of sagemaker can be access using this link <a href="https://docs.AWS.amazon.com/sagemaker/latest/dg/image-classification.html">https://docs.AWS.amazon.com/sagemaker/latest/dg/image-classification.html</a>

There are different type of algorithms are offered by sagemaker, which can be used according to the need. Information of different type of algorithms Sagemaker can be accessed here. https://docs.AWS.amazon.com/sagemaker/latest/dg/algos.html

Writing our model:

First we are saving our S3 bucket and main folder name into variables to access again time to time.

######### Team Members: ######## Muhammad Haseeb Anwar ######## Moeez Ur Rehman ######### Sehrish Kanwal ######### Harmain Haidar # 53 Bucket Name bucket\_name='cloud-computing-dataset' # Our Main Folder inside the S3 bucket Which has subfolders of our classes # One Sub-Folder will be considered as One Class dataset\_name = 'CC-Dataset' print('Name of the bucket is: '+dataset name) print('Name of dataset folder is: '+bucket\_name) Name of the bucket is: CC-Dataset Name of dataset folder is: cloud-computing-dataset

Here we are importing the sagemaker library and setting up environment.

**Get\_execution\_role()** method gives us the role which we are using to run the sagemaker notebook.

session() method is used to get a sagemaker session.

**Image\_uris.retrieve()** method is used for generating ECR image URIs for pre-built SageMaker Docker image, the arguments of the methods can be studied extensively using below link. https://sagemaker.readthedocs.io/en/stable/api/utility/image\_uris.html#sagemaker.image\_uris.retrieve

As we are using prebuilt image classification model of sagemaker we have passed this algorithm name in the perimeter.

```
[13]: #Setting Up Our Environment

# Importing Sagemaker

# getting execution role of notebook

# defining algorithm type in Image_Uri method

import sagemaker

from sagemaker import get_execution_role

from sagemaker.amazon.amazon_estimator import get_image_uri

role = get_execution_role()

session = sagemaker.session()

#sogemaker.image_uris.retrieve

#get_image_uri

image_uris = sagemaker.image_uris.retrieve(region=session.boto_region_name, framework ='image-classification')

print('Region name: '+session.boto_region_name)

print('Algorithm Used: image-classification ')
```

Region name: eu-central-1 Algorithm Used: image-classification There are multiple ways to feed images to the model for training. The SageMaker Image Classification algorithm supports both RecordIO and conventional image formats like JPG and JPEG. In this project we are going to use the RecordIO format for training.

### What is Record IO format?

Data loading is a critical component of any machine learning system. With smaller number of training images, it might not be a problem to used them as they are, but with larger datasets, data loading into training model can become performance critical.

In simple words, Record IO format converts images into binary data exchange formats, RecordIO is efficient data format developed by Apache MXnet it resizes the image into 256 \* 256, then compress into JPEG format. After that, it saves a header that indicates the index and label for that image to be used when constructing the *Data* field for that record. It then pack several images together into a file.

More information of RecordIO file can be read here: https://mxnet.apache.org/versions/1.9.1/api/architecture/note\_data\_loading

Sagemaker recommend storing images as records and packing them together, the major benefit is Storing images in RecordIO format greatly reduces the size of the dataset on the disk.

In below code we specify the path of the script which converts images into RecordIO files

```
BASE_DIRECTORY='/tmp'
%env BASE_DIRECTORY=$BASE_DIRECTORY
%env S3_BUCKET_NAME = $bucket_name
%env DATASET_NAME = $dataset_name
import sys,os
suffix='/mxnet/tools/im2rec.py'
im2rec = list(filter( (lambda x: os.path.isfile(x + suffix )), sys.path))[0] + suffix
%env IM2REC=$im2rec
env: BASE_DIR=/tmp
env: S3_DATA_BUCKET_NAME=cloud-computing-dataset
env: DATASET_NAME=CC-Dataset
env: IM2REC=/home/ec2-user/anaconda3/envs/mxnet_p36/lib/python3.6/site-packages/mxnet/tools/im2rec.py
```

As we have specified the script which transforms our images into Record IO file, we now pull all S3 images.



Now we transform our fetched images into Record IO file, we have kept the training ratio to 70%, while Testing ratio to 30%. The Record IO files will be created in this step with the above ratio.



Now we upload our created RecordIO files back into our S3 bucket, which then be used as an input for training of our model.

	Carl (1999) (1997)			
ucket = buck	et_name			
r <mark>int</mark> (bucket	i.			
raining_path	s3 = 's3://{}/{}/train/'.	format(bucket, dataset_	name)	
alidation_pa	:h_S3 = 's3://{}/{}/validat	tion/'.format(bucket, d	ataset_name)	
rint(trainin	_path_s3)			
orint(validat	ion_path_S3)			
t Delete any	existing data			
aws s3 rm s3	<pre>//{bucket}/{dataset_name},</pre>	/trainrecursive		
aws s3 rm s3	//{bucket}/{dataset_name}	/validationrecursive		
t Upload the	rec files to the train and	validation folders		
aws s3 cp /t	<pre>np/{dataset_name}_train.re</pre>	<pre>\$training_path_s3</pre>		
aws s3 cp /t	<pre>np/{dataset_name}_test.rec</pre>	<pre>\$validation_path_S3</pre>		
loud-computi	ig-dataset			
3://cloud-co	nuting-dataset/CC-Dataset	train/		

The uploaded RecordIO files in our S3 bucket will look like this.

ain/					D 0	opy S3 I
Objects Properties						
Objects (1)						
Objects are the fundamental entities your objects, you'll need to explicitly	stored in Amazon S3. You grant them permissions. L	can use Amazon S3 inven	tory 🛃 to get a list (	of all objects in yo	ur bucket. For others	to access
C 🗇 Copy S3 URI	Copy URL	🕑 Download	Open 🖸	Delete	Actions <b>v</b>	]
Create folder	load				20	
<b>Q</b> Find objects by prefix					< 1	>
Name	▲ Tune ▽	Last modified		V Siza	⊽ Storage	class
Name	- Туре Ф	Last mounied		* 5126	* Storage	e class
CC-Dataset_train.re	ec rec	July 1, 2022, 00:3	4:46 (UTC+02:00) alidation/	8.5	MB Standar	d
CC-Dataset_train.re	ec rec	July 1, 2022, 00:3	4:46 (UTC+02:00) alidation/	8.5	MB Standar	d by S3 UF
CC-Dataset_train.re	ec rec	July 1, 2022, 00:3-	4:46 (UTC+02:00) alidation/	8.5	MB Standar	d by S3 UF
CC-Dataset_train.re	ec rec	July 1, 2022, 00:3-	4:46 (UTC+02:00) alidation/	8.5	MB Standar	d by S3 UF
CC-Dataset_train.re	ec rec	July 1, 2022, 00:3-	4:46 (UTC+02:00) alidation/	8.5	MB Standar	d by S3 UF
CC-Dataset_train.re nazon S3 > Buckets > cloud alidation/ Objects Properties Objects (1)	d-computing-dataset	July 1, 2022, 00:3	4:46 (UTC+02:00) alidation/	8.5	MB Standar	d by S3 UF
CC-Dataset_train.re	ec rec d-computing-dataset stored in Amazon S3. You o grant them permissions. Le	July 1, 2022, 00:3	4:46 (UTC+02:00) alidation/ ory [2] to get a list of	8.5	MB Standar	d Py S3 UF
CC-Dataset_train.re	ec rec d-computing-dataset stored in Amazon S3. You grant them permissions. Le	July 1, 2022, 00:3	4:46 (UTC+02:00) alidation/ ory [2] to get a list of Open [2]	8.5 f all objects in your Delete	MB Standar	d py S3 UF
CC-Dataset_train.re	ec rec d-computing-dataset stored in Amazon S3. You of grant them permissions. Le Copy URL	July 1, 2022, 00:3	4:46 (UTC+02:00) alidation/ ory [2] to get a list of Open [2]	8.5 f all objects in your Delete	MB Standar	d Py S3 UF
CC-Dataset_train.re	ec rec d-computing-dataset stored in Amazon S3. You o grant them permissions. Lo Copy URL load	July 1, 2022, 00:3	4:46 (UTC+02:00) alidation/ ory 🛃 to get a list of Open 🔁	8.5 f all objects in your Delete	MB Standar	d yy S3 UF access
CC-Dataset_train.re	ec rec d-computing-dataset stored in Amazon S3. You grant them permissions. Lo Copy URL load	July 1, 2022, 00:3	4:46 (UTC+02:00) alidation/ ory 🕑 to get a list of Open 🖸	all objects in your	MB Standar	o access

We have now done our preprocessing; the data is ready to be trained. Now are going towards the process of training our model using the created Record IO files.

We are here defining the Record IO paths to the training and validation functions. For information of the inputs. Training Input() method can be found here:

https://sagemaker.readthedocs.io/en/stable/api/utility/inputs.html#sagemaker.inputs.TrainingInput

```
[20]:
      # Documentation of the function sagemaker.inputs.TrainingInput is available here
      # https://sagemaker.readthedocs.io/en/stable/api/utility/inputs.html#sagemaker.inputs.TrainingInput
      # Create a definition for input data used by an SageMaker training job.
      train data = sagemaker.inputs.TrainingInput(
          training_path_s3,
          distribution='FullyReplicated',
          content_type='application/x-recordio',
          s3_data_type='S3Prefix'
      validation data = sagemaker.inputs.TrainingInput(
          validation path S3,
          distribution='FullyReplicated',
          content_type='application/x-recordio',
          s3_data_type='S3Prefix'
      )
      data_channels = { 'train': train_data, 'validation': validation_data}
      print(train_data)
      print(validation_data)
```

<sagemaker.inputs.TrainingInput object at 0x7f813d49e2e8> <sagemaker.inputs.TrainingInput object at 0x7f813d49e320>

Defining the output location of out model, as well as initializing the estimator function. Sagemaker handles endto-end Amazon Sagemaker training and deployment tasks. More documentation can be read https://sagemaker.readthedocs.io/en/stable/api/training/estimators.html

```
[43]: # The are defining the output location for model
s3_output_location = 's3://{}/output'.format(bucket, dataset_name)
# we have used ml.p3.2xlarge isntance for traning
image_classifier = sagemaker.estimator.Estimator(
    role=role,
    image_uri=image_uris,
    instance_count=1,
    instance_type='ml.p3.2xlarge',
    output_path=s3_output_location,
    sagemaker_session=session
)
print('done')
done
```

Image classification Hyperparameters,

we have defined the

image shape as 3,244,244 which is same as the image shape of our RecordIO files.

number of classes which in our case are 4,

Augmentation type here is important as we are taking the color into account, so we have chosen 'crop color'.

Epoch: We have not provided any value for epochs so it will take the default value 30.

**Learning rates:** The learning rate controls how quickly the model is adapted to the problem. Smaller learning rates require more training epochs given the smaller changes made to the weights each update, whereas larger learning rates result in rapid changes and require fewer training epochs. The valid values are between 0 and 1.

more documentation can be found here https://docs.AWS.amazon.com/sagemaker/latest/dg/IC-Hyperparameter.html

```
[58]: num_classes=! ls -l {base_dir}/{dataset_name} | wc -l
      num_classes=int(num_classes[0]) - 3
      num_training_samples=! cat {base_dir}/{dataset_name}_train.lst | wc -1
      num_training_samples = int(num_training_samples[0])
      # Details on Sagemaker built-in Image Classifier hyperparameters
      # available here: https://docs.aws.amazon.com/sagemaker/latest/dg/IC-Hyperparameter.html
      base_hyperparameters=dict(
          use pretrained model=1,
          image_shape='3,224,224',
          num_classes=num_classes,
          augmentation_type='crop_color', #taking corresponding Hue-Saturation-Lightness into account
          num_training_samples=num_training_samples,
      # These are hyperparameters are important which can affect the model training success:
      hyperparameters={
          **base hyperparameters,
          **dict(
             learning_rate=0.001,
              mini_batch_size=5,
          )
      }
      image_classifier.set_hyperparameters(**hyperparameters)
      hyperparameters
      print('No of tranining Samples: '+str(num_training_samples))
      print ('No of Classes: '+str(num_classes))
      No of tranining Samples: 464
```

No of Classes: 4

Now starting our training job, we have given all our parameters as an input to our training job.

The training job is started and will provide the path of the model where it will be stored.

```
[*]: %%time
```

import time now = str(int(time.time())) training\_job\_name = 'IC-' + dataset\_name.replace('\_', '-') + '-' + now

image\_classifier.fit(inputs=data\_channels, job\_name=training\_job\_name, logs=True)

job = image\_classifier.latest\_training\_job model\_path = f"{BASE\_DIR}/{job.name}"

print(f"\n\n Finished training! The model is available for download at: {image\_classifier.output\_path}/{job.name}/output/model.tar.gz")

2022-06-30 23:08:48 Starting - Starting the training job...ProfilerReport-1656630528: InProgress

2022-06-30 23:09:40 Starting - Preparing the instances for training...... 2022-06-30 23:10:43 Downloading - Downloading input data..

1

2022-06-30 23:13:04 Training - Training image download completed. Training in progress.[23:13:11] /opt/brazil-pkg-cach	e/packages/AIAlgorithm
sMXNet/AIAlgorithmsMXNet-1.3.x_ecl_Cuda_10.1.x.11282.0/AL2_x86_64/generic-flavor/src/src/operator/nn/./cudnn/./cudnn_a	lgoreg-inl.h:97: Runni
ng performance tests to find the best convolution algorithm, this can take a while (setting env variable MXNET_CUDN	IN_AUTOTUNE_DEFAULT to
0 to disable)	
[06/30/2022 23:13:15 INFO 140493106341696] Epoch[0] Batch [20]#011Speed: 23.524 samples/sec#011accuracy=0.428571	
[06/30/2022 23:13:16 INFO 140493106341696] Epoch[0] Batch [40]#011Speed: 32.955 samples/sec#011accuracy=0.609756	
[06/30/2022 23:13:18 INFO 140493106341696] Epoch[0] Batch [60]#011Speed: 38.072 samples/sec#011accuracy=0.708197	
[06/30/2022 23:13:20 INFO 140493106341696] Epoch[0] Batch [80]#011Speed: 41.102 samples/sec#011accuracy=0.767901	
[06/30/2022 23:13:21 INFO 140493106341696] Epoch[0] Train-accuracy=0.784783	
[06/30/2022 23:13:21 INFO 140493106341696] Epoch[0] Time cost=10.725	
[06/30/2022 23:13:22 INFO 140493106341696] Epoch[0] Validation-accuracy=0.995000	
[06/30/2022 23:13:23 INFO 140493106341696] Storing the best model with validation accuracy: 0.995000	
[06/30/2022 23:13:23 INFO 140493106341696] Saved checkpoint to "/opt/ml/model/image-classification-0001.params"	
[06/30/2022 23:13:25 INFO 140493106341696] Epoch[1] Batch [20]#011Speed: 55.178 samples/sec#011accuracy=0.895238	
[06/30/2022 23:13:27 INFO 140493106341696] Epoch[1] Batch [40]#011Speed: 55.481 samples/sec#011accuracy=0.902439	
[06/30/2022 23:13:28 INFO 140493106341696] Epoch[1] Batch [60]#011Speed: 55.671 samples/sec#011accuracy=0.927869	
[06/30/2022 23:13:30 INFO 140493106341696] Epoch[1] Batch [80]#011Speed: 55.449 samples/sec#011accuracy=0.938272	
[06/30/2022 23:13:31 INFO 140493106341696] Epoch[1] Train-accuracy=0.936957	
[06/30/2022 23:13:31 INFO 140493106341696] Epoch[1] Time cost=8.199	
[06/30/2022 23:13:32 INFO 140493106341696] Epoch[1] Validation-accuracy=1.000000	
[06/30/2022 23:13:33 INFO 140493106341696] Storing the best model with validation accuracy: 1.000000	
[06/30/2022 23:13:33 INFO 140493106341696] Saved checkpoint to "/opt/ml/model/image-classification-0002.params"	
[06/30/2022 23:13:35 INFO 140493106341696] Epoch[2] Batch [20]#011Speed: 54.774 samples/sec#011accuracy=0.914286	
[06/30/2022 23:13:37 INFO 140493106341696] Epoch[2] Batch [40]#011Speed: 55.316 samples/sec#011accuracy=0.951220	
[06/30/2022 23:13:39 INFO 140493106341696] Epoch[2] Batch [60]#011Speed: 55.483 samples/sec#011accuracy=0.963934	
[06/30/2022 23:13:40 INFO 140493106341696] Epoch[2] Batch [80]#011Speed: 55.151 samples/sec#011accuracy=0.967901	
[06/30/2022 23:13:41 INFO 140493106341696] Epoch[2] Train-accuracy=0.971739	
[06/30/2022 23:13:41 INFO 140493106341696] Epoch[2] Time cost=8.231	
[06/30/2022 23:13:42 INFO 140493106341696] Epoch[2] Validation-accuracy=1.000000	
[06/30/2022 23:13:45 INFO 140493106341696] Epoch[3] Batch [20]#011Speed: 53.828 samples/sec#011accuracy=0.990476	
[06/30/2022 23:13:47 INFO 140493106341696] Epoch[3] Batch [40]#011Speed: 55.007 samples/sec#011accuracy=0.990244	
[06/30/2022 23:13:48 INFO 140493106341696] Epoch[3] Batch [60]#011Speed: 55.255 samples/sec#011accuracy=0.977049	
[06/30/2022 23:13:50 INFO 140493106341696] Epoch[3] Batch [80]#011Speed: 55.150 samples/sec#011accuracy=0.980247	
[06/30/2022 23:13:51 INFO 140493106341696] Epoch[3] Train-accuracy=0.976087	
[06/30/2022 23:13:51 INFO 140493106341696] Epoch[3] Time cost=8.235	

print(f"\n\n Finished training! The model is available for download at: {image\_classifier.output\_path}/{job.name}/output/model.tar.gz")

Finished training! The model is available for download at: s3://cloud-computing-dataset/CC-Dataset/output/IC-CC-Dataset-1656630528/output/m odel.tar.gd

We can also see our S3 bucket where the model is saved.

operties					
damental entities st	ored in Amazon S3. You	can use Amazon S3 inven	tory 🛃 to get a list o	of all objects in yo	our bucket. For others to acces
need to explicitly gr	ant them permissions. L	earn more 🗹			
opy S3 URI	🗇 Copy URL	🕑 Download	Open 🖸	Delete	Actions 🔻
T Uplo	ad				
		- 1			
					< 1 >
by prefix					
	amental entities st need to explicitly gr opy S3 URI	amental entities stored in Amazon S3. You teed to explicitly grant them permissions. L opy S3 URI	amental entities stored in Amazon S3. You can use Amazon S3 inven need to explicitly grant them permissions. Learn more opy S3 URI Copy URL Download	amental entities stored in Amazon S3. You can use Amazon S3 inventory 🖓 to get a list of the explicitly grant them permissions. Learn more 🖓 Opy S3 URI 🖸 Copy URL 🔛 Download Open 🖓 Open 🖓	amental entities stored in Amazon S3. You can use Amazon S3 inventory [2] to get a list of all objects in you need to explicitly grant them permissions. Learn more [2] opy S3 URI [] Copy URL [] Download [] Open [2] [] Delete [] Upload

## 3.1.5 Deploying endpoint on sagemaker

As our model is trained, we now have to deploy our endpoint, which then will be used for our predictions.

```
[63]: %%time
# Deploying our trained model to an endpoint which will then we used by our app to predict the currency
deployed_endpoint = image_classifier.deploy(
    initial_instance_count = 1,
    instance_type = 'ml.t2.medium'
)
-------!CPU times: user 141 ms, sys: 16.2 ms, total: 157 ms
Wall time: 4min 31s
[64]: print(deployed_endpoint)
```

<sagemaker.predictor.Predictor object at 0x7f813cab0d30>

The deployed endpoint is available and in service now.

Now there are many ways to call this endpoint, we can predict our images from this notebook or we can create a webapp which will call our endpoint using Api Gateway and Lambda functions and will show our predictions, first we are showing how this can be done using notebook instance.

Create a function which invokes the endpoint and returns the result of prediction.

$\leftarrow \rightarrow $ C C	) 🔒 ≅ https://	/eu-central-1.console.aws. <b>amazon.c</b>	om/sagem	aker/home?region=eu-central-1#/endpoints				5	$\bigtriangledown$ $\checkmark$	≡
aws Services Q Search	for services, featur	res, blogs, docs, and more		[Alt+S]		<u>ک</u>	0	Frankfurt 🔻	MuhammadHaseebAn	war 🔻
Git repositories	O We wa	nt to hear from you. Take our sho	rt 5-10 mi	nute survey and help us make SageMaker better fo	or customers like you.				Take Survey	×
Processing	Amazo	n SageMaker > Endpoints								
Training	Ene	dpoints			C Updat	te endpoint	Ac	tions 🔻	Create endpoint	
▼ Inference		Search endpoints							< 1 > @	
Compilation jobs		, search chaponno								
Marketplace model packages		Name	$\nabla$	ARN		Creation time	•	Status $\triangledown$	Last updated	
Models				annunurus a annahan au santral						
Endpoint configurations	0	image-classification-		1:540180785457:endpoint/image-classification	n-	Jun 30, 2022	2	⊘ InService	Jun 30, 2022	
Endpoints		2022-00-30-23-23-23-030		2022-06-30-23-23-29-838		23.23 010			23.27 010	
Batch transform jobs										
▼ Edge Manager										
Get started										
Edge packaging jobs										
Edge device fleets										

### 3.1.6 Testing the model from notebook

```
ar the exercise the exercise provides and the exercise
       [66]: # If we want to check out model' prediction through this notebook instance
      # we will create a function which will call our endpoint here and return the model prediction
      # we will have to upload some test images to our s3 bucket which this method will use
      import json
      import numpy as np
      import os
      def classify_deployed(file_name, classes):
          payload = None
          with open(file_name, 'rb') as f:
             payload = f.read()
              payload = bytearray(payload)
          result = deployed_endpoint.predict(payload, initial_args={'ContentType': 'image/jpeg'})
          #result = json.loads(deployed_endpoint.predict(payload))
          #result = deployed_endpoint.predict(payload)
          #best_prob_index = np.argmax(result)
          #return (classes[best_prob_index], result[best_prob_index])
          resultarray = (result.decode('UTF-8')[1:len(result)-1]).split(",")
          for i in range(len(classes));
             print(classes[i] + ":" + str(resultarray[i]))
          return result
      print("Function created")
      Function created
```

Now upload a sample image into S3 Bucket, We have used a 50 euro image and uploaded in into the folder test\_images on S3.

store your archive datasets in the lo	w-cost Amazon 55 Gi	lacier storage classes.				view tutoria
Amazon S3 > Buckets > cloud-	computing-dataset	> CC-Dataset/ > Test_ima	ages/			
<b>T</b>						D Copy S3 U
Test_Images/						
Objects Properties						
Objects (1)						
Objects (1) Objects are the fundamental entities st	ored in Amazon S3. You c	can use Amazon S3 inventory 🔽 t	to get a list of all objects in	vour bucket. For others	to access your objects, yo	ou'll need to explicitly grant
Objects (1) Objects are the fundamental entities st them permissions. Learn more 🛂	ored in Amazon S3. You o	can use Amazon S3 inventory 🗹 t	to get a list of all objects in	your bucket. For others	to access your objects, yo	ou'll need to explicitly grant
Objects are the fundamental entities st them permissions. Learn more C C C Copy S3 URI	ored in Amazon S3. You o	can use Amazon 53 inventory [2] 1 단 Download Ope	to get a list of all objects in the second	your bucket. For others	to access your objects, yo	ou'll need to explicitly grant
Objects (1) Objects are the fundamental entities st them permissions. Learn more [2] [C] [] Copy S3 URI [Q, Find objects by prefix	ored in Amazon S3. You o	can use <b>Amazon 53 inventory [2</b> ] t 단 Download Opt	to get a list of all objects in en 🛛 Delete	your bucket. For others	to access your objects, yo	Upload
Objects (1) Objects are the fundamental entities st them permissions. Learn more C C C C C C C C C C C C C C C C C C C	ored in Amazon S3. You o	can use Amazon 53 inventory [2] t [1] Download Opt	to get a list of all objects in en 🛛 Delete	your bucket. For others	to access your objects, yo Create folder Size	Storage class
Objects (1) Objects are the fundamental entities st them permissions. Learn more [2] C ① Copy S3 URI Q. Find objects by prefix Name S0-euro.JPG	Copy URL	can use Amazon 53 inventory [2] t [2] Download Ope □ □ □ Last modified July 1, 2022, 01:38	to get a list of all objects in en  Delete :08 (UTC+02:00)	Vour bucket. For others	to access your objects, yo Create folder Size マ 22.0 KB	Storage class

In this step we are getting our image from S3 and saving them into a variable called Euro50.



Now we are calling our classify\_deployed function to predict our image, and the result is shown in below image.

```
43... connectea.
      HTTP request sent, awaiting response... 200 OK
      Length: 22513 (22K) [image/jpeg]
      Saving to: 'test.jpg'
                         100%[======>] 21.99K --.-KB/s in 0.001s
      test.jpg
      2022-06-30 23:39:44 (21.8 MB/s) - 'test.jpg' saved [22513/22513]
[68]:
[70]: object_categories = [
          "EURO-10",
          "EURO-5",
          "EURO-20",
          "EURO-50"
      1
      output = classify_deployed(Euro50,object_categories)
      EURO-10:2.640426464495249e-05
      EURO-5: 4.2157054849667475e-05
      EURO-20: 9.781115659279749e-05
      EURO-50: 0.9998335838317871
```

As we can see the EURO-50 image has the highest prediction value, our model is 99.9% confident that the provided image in of 50 Euro note.

# 3.1.7 Cleanup Sagemaker

**Important:** Important is to stop the notebook after you have done creating the model, otherwise AWS will keep charging you for the time the notebook is in service.

lote	ebook instances		C	Actions 🔺 🛛 C	eate notebook instance
Q	Search notebook instances			Open Jupyter	< 1 > @
	Name 🗢	Instance	Creation time	Stop	Actions
þ	Cloud-Computing-Project-Image- Recognition-NB-Instance	ml.t2.medium	May 29, 2022 13:35 UTC	Start Update settings	Open Jupyter   Open JupyterLab
				Add/Edit tags	

# 4. Web Application:

The web application is created for demo purposes of the model deployed on AWS Sagemaker. It is created using React libraries in addition to using amplify library which streamlines the connection of the web application with the AWS cloud setup.

For setting up amplify in the project we need to install amplify and with in the project directory execute the command:

amplify init

Now we need to setup an API endpoint on the AWS API Gateway. We use the command:

amplify add api

We further follow the steps in the process executed by the command to make a POST Rest api. When the api end point is created locally we push the setup on the AWS using:

amplify push

The web application consists of 2 main components:

Image Capture:

This component uses the camera of the device to capture the image to be identified.

```
JS ImageCapture.js ∪ ×
src > components > Js ImageCapture.js > 😤 ImageCapture > 😚 render
             // Moeez comments: This component opens and captures or screenshots
 26
 27
             // the image in the camera session which is then used for image recognition
 28
             return (
               <div>
 29
                 <div>
 30
                    <Webcam
 31
                      audio={false}
 32
                      height={IMAGE HEIGHT}
 33
 34
                      width={IMAGE WIDTH}
                      ref={this.setRef}
 35
                      screenshotFormat="image/jpeg"
 36
                      screenshotWidth={IMAGE WIDTH}
 37
                      videoConstraints={videoConstraints}
 38
 39
                    />
                 </div>
 40
 41
                 <Form.Button onClick={this.handleCapture}>Classify</Form.Button>
 42
               </div>
 43
 44
             );
 45
           }
 46
```

Classified Image:

This component displays the result in a format of a card with header, information and details.

```
JS ClassifiedImage.js ∪ ×
      src > components > _s ClassifiedImage.js > ...
       36
                 // Moeez comments: This component creates a React UI Card which is consists of a
App
       37
       38
                 // Header, Meta and a Description element.
       39
                 render() {
       40
                   return (
                      <Card style={{width: '224px'}}>
       41
                       <Image <pre>src={this.props.imageSrc} />
       42
                        <Card.Content>
       43
                          <Card.Header>
       44
                            { this.state.bestLabel ? this.state.bestLabel : "Loading..." }
       45
       46
                          </Card.Header>
       47
                          <Card.Meta>
                           { this.state.bestLabelScore ? this.state.bestLabelScore : "" }
       48
                          </Card.Meta>
       49
                          <Card.Description>
       50
                           <Accordion defaultActiveIndex={-1} panels={this.accordionPanels()} />
       51
       52
                          </Card.Description>
       53
                       </Card.Content>
                      </Card>
       54
       55
                 }
       56
       57
               }
       58
```

Class:

The main class of the web app is the App.js class where everything is put together.

First we make a form t take the inputs where we want to send the POST request to. The input includes name of the AWS Inference endpoint of Sagemaker, its region and the labels or categories we want to identify or find results of.



Secondly after capturing image and issuing a POST request to AWS API Gateway endpoint, we receive the response and we display it to the user via a Card Group with our component of ClassifiedImage.



snapshot of the request that is sent is attached here:

```
JS App.js M ×
src > Js App.js > 😤 App > 🔗 render
 83
        // Moeez comments: This function calls the AWS API Gateway endpointName
 84
 85
        // and returns the categories with their predictions as the result
 86
         classifier = async (imageSrc) => {
           const base64Image = new Buffer(imageSrc.replace(/^data:image\/\w+;base64,/, ""), 'base64')
 87
 88
           const { predictions } = await API.post(
 89
            aws exports.aws cloud logic custom[0].name,
 90
             '/classify',
 91
             {
              body: {
 92
                 base64Image,
 93
                 endpointName: this.state.endpointName,
 94
                 endpointRegion: this.state.endpointRegion,
 95
 96
              },
            }
 97
 98
           );
           const topProbIndex = argMax(predictions);
 99
           const labels = [].concat(this.state.labels.split(' '));
100
101
           labels.sort();
102
          return {
            labels: labels, predictions, topProbIndex: topProbIndex
103
104
          }
105
         }
106
```

#### 5. Predicting currency note from webapp

**Results:** 

The webpage needs 3 inputs, Sagemaker endpoint name, Region Name and all the classes names with 'space'



We capture images by pressing the classify button, and then prediction score will be shown below.



### We tried to predict the notes with different angles, to see how our prediction is working.

				1	• 🖻 🛧 🚺 🍺 🚥 🗟 🗯
Clear Images	10	20	50		5
EURO-5 0.9995836615562439 • Show Score Details	EURO-10 0.9951698184013367 • Show Score Details	EURO-20 0.883262038230896 • Show Score Details	EURO-50 0.9913842678070068 • Show Score Details	EURO-5 0.9993664622306824 • Show Score Details	EURO-5 0.999526858329773 • Show Score Details
	5 7 7 7 7 7 7 7 7 7				5.00
EURO-5 0.9973311424255371 • Show Score Details	EURO-5 0.9894456267356873 • Show Score Details	EURO-5 0.994642972946167 • Show Score Details	EURO-5 0.8661041855812073 • Show Score Details	EURO-5 0.9968541264533997 • Show Score Details	EURO-5 0.9250454306602478 • Show Score Details