AI with Flutter

Rishit Dagli

About Me

Rishit Dagli

Rishit-dagli

rishit_dagli

Rishit Dagli

rishitdagli.ml

hello@rishitdagli.ml

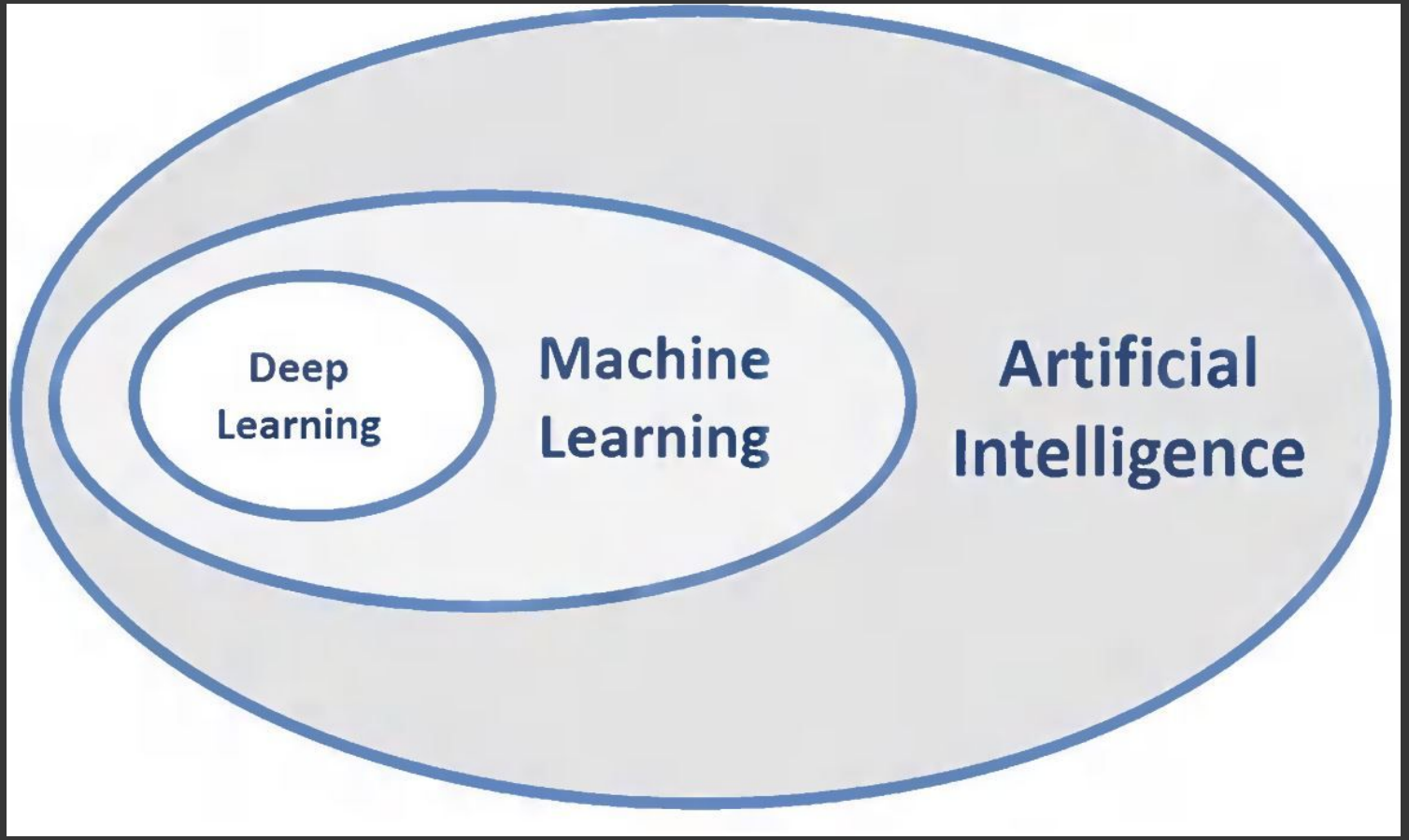
@rishit.dagli

rishit-dagli

ML, DL?

ML - capability to learn without being explicitly programmed

DL - Learning based on DNNs



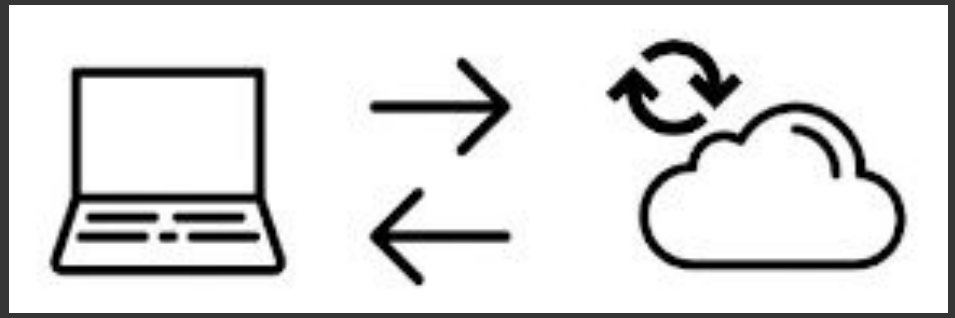
Flutter

Edge?

- Local or near Local processing
- Not just anywhere in cloud
- Low latency
- No network availability
- Best for real time decision making

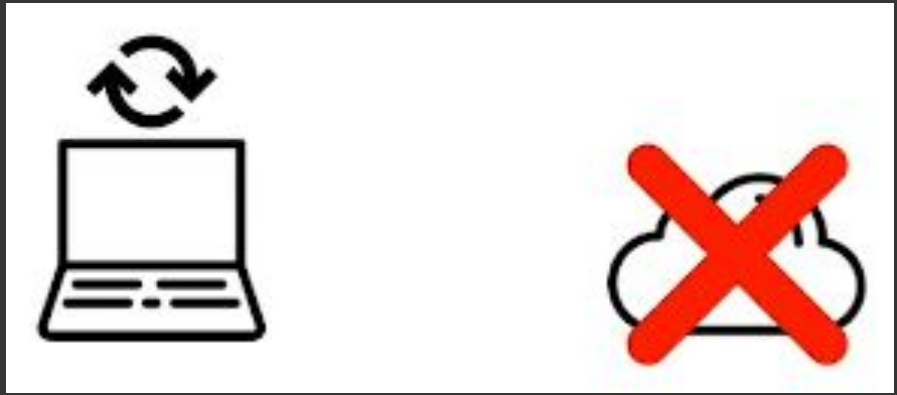
Cloud vs Edge - Cloud?

- Get data locally
- Send it to cloud
- Process it in cloud
- Send response back



Cloud vs Edge - Edge?

- No need to send to cloud
- Secure
- Less impact on network



Flutter

Cloud vs Edge?

- Does not mean no cloud whatsoever
- Cloud can be used for training of models
- Inference is on edge

Why care about edge?

Network

- Expensive in cost, bandwidth or power
- Can be impossible
- Sending audio/ video is data intensive

Why care about edge?

Latency

- Can’t handle latency

Why care about edge?

Security

- Personal data
- Data is sensitive

How to do edge deployment?

- Can I use a normal model?
- Shrink the model
- Use a .tflite model
- Or some pretrained models too

Convert models

```
converter = tf.lite.TFLiteConverter.from_saved_model(export_dir)
tflite_model = converter.convert()
```

Convert models

- Get the TFLite model
- You can use the package tflite
- You cannot use pre trained models with it
- Save the model in Firebase

ML Kit

- Part of Firebase
- Allows you to use pre trained models
- And custom TF models too

ML Kit

- Part of Firebase
- Allows you to use pre trained models
- Easily use barcode scanning, text, landmark, label detectors
- And custom TF models too

Make a face recognizer on edge

Add dependencies

- image_picker
- firebase_ml_vision

Add them to pubsec.yaml

```
dependencies:
  flutter:
    sdk: flutter
  image_picker: ^0.6.1+4
  firebase_ml_vision: ^0.9.2+1
```

Create Firebase Project

Create a simple Scaffold

```
class FacePage extends StatefulWidget {
  @override
  _FacePageState createState() => _FacePageState();
}

class _FacePageState extends State<FacePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Face Detector')
      ),
      body: Container(),
    );
}
```

Create a simple Scaffold

```
class FacePage extends StatefulWidget {
  @override
  _FacePageState createState() => _FacePageState();
}
```

To track the selected images and any faces detected

Add a FAB

```
floatingActionButton: FloatingActionButton(
        onPressed: (){},
        tooltip: 'Pick Image',
         child: Icon(Icons.add_a_photo)
```

- **You made a FAB with a suitable icon**

WorkFlow

- Select an image
- Load image for processing
- Perform face detection

WorkFlow

- Select an image
  - Load image for processing
  - Perform face detection

Selecting images

- Very easy with the image_picker

```
final imageFile = await ImagePicker.pickImage(
  source: ImageSource.gallery,
);
```

- Can also use camera

Selecting images

```
final imageFile = await ImagePicker.pickImage(
  source: ImageSource.gallery,
);
```

Get output of this asynchronous function

WorkFlow

- Select an image
- Load image for processing
- Perform face detection

Loading images

```
final imageFile = await ImagePicker.pickImage(
        source: ImageSource.gallery);

final image = FirebaseVisionImage.fromFile(imageFile);
```

Loads and store image in a format suitable for feature detection

WorkFlow

- Select an image
- Load image for processing
- Perform face detection

Face detection

```
final faceDetector =
FirebaseVision.instance.faceDetector();
```

- Instance of `FirebaseVision` class
- Initialize it with a `faceDetector()`

Face detection - continue

```
FirebaseVision.instance.faceDetector();
FirebaseVision.instance.barcodeDetector();
FirebaseVision.instance.labelDetector();
FirebaseVision.instance.textDetector();
FirebaseVision.instance.cloudLabelDetector();
```

- Other available detectors

Face detection - continue

- Also provide optional parameters
- Control accuracy
- Speed
- Look for ears, eyes, nose

Cheat Sheet

Settings	
Performance mode	`FAST` (default) | `ACCURATE` Favor speed or accuracy when detecting faces.
Detect landmarks	`NO_LANDMARKS` (default) | `ALL_LANDMARKS` Whether to attempt to identify facial "landmarks": eyes, ears, nose, cheeks, mouth, and so on.
Detect contours	`NO_CONTOURS` (default) | `ALL_CONTOURS` Whether to detect the contours of facial features. Contours are detected for only the most prominent face in an image.
Classify faces	`NO_CLASSIFICATIONS` (default) | `ALL_CLASSIFICATIONS` Whether or not to classify faces into categories such as "smiling", and "eyes open".
Minimum face size	`float` (default: `0.1f`) The minimum size, relative to the image, of faces to detect.
Enable face tracking	`false` (default) | `true` Whether or not to assign faces an ID, which can be used to track faces across images. Note that when contour detection is enabled, only one face is detected, so face tracking doesn't produce useful results. For this reason, and to improve detection speed, don't enable both contour detection and face tracking.

Face detection - continue

```
final faceDetector = FirebaseVision.instance.faceDetector(
    FaceDetectorOptions(
      mode: FaceDetectorMode.fast,
      enableLandmarks: true,
)
```

Face detection - continue

- Now asynchronously pass your image
- And get a list of face coordinates
- And any other parameters

Face detection - continue

```
class Face{
  final Rectangle<int> boundingBox;
  final double headEulerAngleY;
  final double headEulerAngleZ;
  final double leftEyeOpenProbability;
  final double rightEyeOpenProbability;
  final double smilingProbability;
  final int trackingId;
  FaceLadmark getLandmark(
    FaceLandmarkType landmark,
  ) => _landmarks[landmark]
```

Face detection - continue

- So I will just add all of this in a function
- Make a call to setState
- Mark the state as updated once face detection is complete
- Add this function in the FAB on pressed we talked about earlier

A problem?

- We have 2 await-
  - Getting image from user
  - Getting faces list
- Face page widget might not be active or mounted

A problem?

- Can close the app
- Or navigate away from page
- Wasted processing or battery
- Solution to this?

mounted

- Check if widget is mounted
- Before setting the state

```
if (mounted) {
    setState(() {
      _imageFile = imageFile;
      _faces = faces;
    });
```

What now?

- I have my image
- Faces list with coordinnates
- Faces are detected properly

- But I still need to modify the image itself to show a bounding box over face
- Any guesses, what should I use?

CustomPaint widget

- Gives you a canvas
- Allows you to draw almost anything
- Execute simple paint commands

CustomPaint widget

```
final customPaint = CustomPaint(
  painter: myPainter(),
  child: AnyWidget(),
)

class MyPainter extends CustomPainter{
  @override
  void paint(ui.Canvas canvas, ui.Size size){
    // implement paint
  }
  @override
  void shouldRepaint(CustomPainter oldDelegate){
    // implement shoudRepaint
  }
}
```

CustomPaint widget

```
class MyPainter extends CustomPainter{
  @override
  void paint(ui.Canvas canvas, ui.Size size){
    // implement paint
  }
  @override
  void shouldRepaint(CustomPainter oldDelegate){
    // implement shoudRepaint
  }
}
```

Real drawing takes place

CustomPaint widget

```
class MyPainter extends CustomPainter{
  @override
  void paint(ui.Canvas canvas, ui.Size size){
    // implement paint
   canvas.drawCircle(Offset.zero, Offset(50, 50), 20, Paint())
}
```

Draw figures easily

CustomPaint widget

```
final Paint paint = Paint()
    ..style = PaintingStyle.stroke
    ..strokeWidth = 15.0
    ..color = Colors.blue;
```

Easily customise properties

CustomPaint widget

```
@override
void shouldRepaint(CustomPainter oldDelegate) => false;
```

- Controls when painter should redraw
- No mutable properties for customPainter
- Return false
- We cannot change what is drawn

Paint over images

```
Future<ui.Image> _loadImage(File file) async {
  final data = await file.readAsBytes();
  return await decodeImageFromList(data);
}
```

- Load image in a format Canvas can understand

Paint over images

```
Future<ui.Image> _loadImage(File file) async {
  final data = await file.readAsBytes();
  return await decodeImageFromList(data);
}
```

Load image as array of raw bytes

Paint over images

```
Future<ui.Image> _loadImage(File file) async {
  final data = await file.readAsBytes();
  return await decodeImageFromList(data);
}
```

Decode image using a Flutter's function

Paint over images

- Create a custom painter
- Pass it our decoded image
- Pass Coordinates for face

Paint over images

```
class FacePainter extends CustomPainter{
  FacePainter(this.image, this.faces);
  final ui.Image image;
  final List<Rect> faces;

  @override
  void paint(ui.Canvas canvas, ui.Size size){}
  @override
  void shouldRepaint(CustomPainter oldDelegate){
    return null;
  }
```

Paint over images

```
@override
 void paint(ui.Canvas canvas, ui.Size size) {
 canvas.drawImage(image, Offset.zero, Paint());
   for (var i = 0; i < faces.length; i++) {
     canvas.drawRect(rects[i], paint);
   }
 }
```

Draw the original image

Paint over images

```
@override
 void paint(ui.Canvas canvas, ui.Size size) {
 canvas.drawImage(image, Offset.zero, Paint());
   for (var i = 0; i < faces.length; i++) {
     canvas.drawRect(rects[i], paint);
   }
 }
```

Draw an unfilled rectangle on each face

Paint over images

```
@override
 void paint(ui.Canvas canvas, ui.Size size) {
 canvas.drawImage(image, Offset.zero, Paint());
   for (var i = 0; i < faces.length; i++) {
     canvas.drawRect(rects[i], paint);
   }
 }
```

Draw an unfilled rectangle on each face

Paint over images

```
@override
  bool shouldRepaint(FacePainter oldDelegate) {
    return image != oldDelegate.image || faces != oldDelegate.faces;
  }
```

Repaint if image or list of faces change

Paint over images

- Reference it from a custom paint widget

```
final facePaint = FacePaint(
painter: myPainter());
```

- Are we done?

Still a problem

- Flutter is not good when you try to draw something out of canvas
- Image can go outside of your canvas

```
SizedBox(
        width: _image.width.toDouble(),
        height: _image.height.toDouble(),
        child: CustomPaint(
          painter: FacePainter(_image, _faces),
        ),
```

Still a problem

- Placing custom paint in a container
- And trying to size it
- Bad idea!!!
- Use another widget

FittedBox

- Fit and scale sizeBox inside it
- Allows other widgets to constrain dimensions

```
FittedBox(
        child: SizedBox(
          width: _image.width.toDouble(),
          height: _image.height.toDouble(),
          child: CustomPaint(
            painter: FacePainter(_image, _faces),
          ),
```

GitHub

github.com/Rishit-dagli/Face-Recognition_Flutter

Flutter

Demo

About Me

Rishit Dagli

Rishit-dagli

rishit_dagli

Rishit Dagli

rishitdagli.ml

hello@rishitdagli.ml

@rishit.dagli

rishit-dagli

Q & A

Thank You

PS: All the code is available on my GitHub

Flutter