



الجمهورية العربية السورية

جامعة دمشق

كلية الهندسة المعلوماتية

## Smart Driver Alert System

(نظام تنبيه السائق الذكي)

تقرير إنجاز المشروع /2/ في الهندسة المعلوماتية

ولاء توما

برزاني برمجة

مرام جعمور

نوران جوخدار

بإشراف: د. ندى غنيم

## الملخص

تم إعداد هذا التقرير لإيضاح الأسس التي بُني عليها هذا المشروع.

الهدف الرئيسي من هذا التقرير هو توثيق عملية تخطيط وتصميم وتطوير نظام برمجي ذكي لتنبيه السائق في حالة النعاس وفي حالة اكتشاف لافتات مرورية لم ينتبه لها السائق او لم يعرف معناها .

بني هذا النظام على فكرتين أساسيتين تخدمان حالات الاستخدام المراد تطبيقها وهم :

1- اكتشاف نعاس السائق من خلال اكتشاف الوجه أولاً Face detection ومن ثم تحديد نقاط معالم الوجه Facial landmarks points وحساب ال EAR & MAR .

2- استخدام تقنية الكشف عن الكائنات Object Detection لاكتشاف اللافتات المرورية في الصورة (او الفيديو) وتصنيفها وتنبيه السائق بها.

تم بناء النظام ك تطبيق للهاتف Mobile App لتسهيل وصول المستخدمين لها كما تم بناء نسخة desktop لتكون جاهزة في حالة استخدامها على ال mini computers مثل ال Raspberry pi

الكلمات المفتاحية: نظام تنبيه ذكي , اكتشاف النعاس , اكتشاف الكائنات , اكتشاف اللافتات المرورية

# جدول المحتويات

2	الملخص
3	جدول المحتويات
4	الفصل الأول: مقدمة
7	الفصل الثاني: الدراسة المرجعية
11	لفصل الثالث: دراسة المتطلبات
11	متطلبات النظام
11	قائمة المتطلبات الوظيفية
11	متطلبات أخرى للنظام (غير وظيفية)
12	توصيف المتطلبات الوظيفية
12	أصحاب المنفعة
12	مخطط حالات الاستخدام
13	توصيف واجهة الاستخدام
14	الفصل الرابع: الموارد والتقنيات المستخدمة:
17	الفصل الخامس: اكتشاف النعاس
28	الفصل السادس: اكتشاف اللافتات المرورية
42	الفصل السابع: الخاتمة
43	المراجع

## الفصل الأول: مقدمة

### مقدمة عامة :

### الذكاء الاصطناعي :

هو فرع من علم الحاسوب. تتعرف الكثير من المؤلفات الذكاء الاصطناعي، على أنه "دراسة وتصميم العملاء الأذكاء"، والعميل الذي هو نظام يستوعب بيئته ويتخذ المواقف التي تزيد من فرصه في النجاح في تحقيق مهمته أو مهمة فريقه. هذا التعريف، من حيث الأهداف والأفعال والتصور والبيئة يرجع إلى Norvig & Russel وتشمل أيضا التعريفات الأخرى المعرفة والتعلم كمعايير إضافية. صاغ عالم الحاسوب جون مكارثي هذا المصطلح بالأساس في عام 1956 وعرفه بنفسه بأنه "علم وهندسة صنع الآلات الذكية". ويعرف أندرياس كابلان ومايكل هاينلين الذكاء الاصطناعي بأنه "قدرة النظام على تفسير البيانات الخارجية بشكل صحيح، والتعلم من هذه البيانات، واستخدام تلك المعرفة لتحقيق أهداف ومهام محددة من خلال التكيف المرن.

### المجالات الفرعية للذكاء الاصطناعي :

يتكون الذكاء الاصطناعي الآن من العديد من المجالات الفرعية ، باستخدام مجموعة متنوعة من التقنيات ، مثل:

الشبكات العصبية ( Neural Networks ) : على سبيل المثال نمذجة الدماغ ، والتنبؤ بالسلسلة الزمنية ، والتصنيف .

الحساب التطوري ( Computation Evolutionary ) : على سبيل المثال الخوارزميات الجينية والبرمجة الجينية .

الروبوتات ( Robots ) : على سبيل المثال التحكم الذاتي ، الاستكشاف الذاتي .

أنظمة قواعد المعرفة ( System Base Knowledge ) : على سبيل المثال أنظمة دعم القرار ، وأنظمة التدريس .

معالجة اللغة الطبيعية (NLP) : على سبيل المثال الترجمة وتوقع النص Smart Agents , تحليل النصوص لاكتشاف معانيها.

الرؤية الحاسوبية (Computer Vision) : هي إحدى مجالات علم الحاسوب، تهدف إلى بناء تطبيقات ذكية قادرة على فهم محتوى الصور كما يفهمها الإنسان. حيث من الممكن أن تأخذ بيانات الصور عدة أشكال كالصور المتعاقبة (فيديو) والمشاهد من عدة كاميرات، بيانات ذات عدة أبعاد مأخوذة من جهاز تصوير طبي. قد أصبحت الرؤية الحاسوبية واحدة من أهم مجالات البحث في عالم الذكاء الاصطناعي. علاوة على ذلك ، فقد أصبح جزءاً من حياتنا الشخصية. نستخدم جميعاً ميزات متنوعة لها تقنيات رؤية الكمبيوتر تعمل في الخلفية. على سبيل المثال ، نستخدم ميزة "فتح القفل بالوجه" في هواتفنا الذكية.

"الرؤية الحاسوبية هي أداة تساعد على اتخاذ قرارات مفيدة بشأن الأشياء المادية الحقيقية والمشاهد بناءً على الصور المحسوسة"

(Sockman & Shapiro 2001)

## مقدمة عن المشروع :

قيادة المركبات أو القيادة أو السياقة هي عملية التحكم في مركبة، كالسيارة والشاحنة والحافلة. يجب على السائق الالتزام بقوانين السير والعديد من المشتتات التي قد تواجهه خلال عملية السياقة و هي عملية مرهقة وقد تسبب العديد من الحوادث بسبب ظاهرة كنعاس السائق او عدم الانتباه او فهم اللافتات المرورية التي قد تصادف السائق في الطريق و مع تطور التكنولوجيا وظهور الكثير من التقنيات التي تساعد الإنسان في حياته اليومية و مع صعود مجال الذكاء الاصطناعي على جميع المستويات وما تقدمه من خدمات كانت مستحيلة في السابق كما تحدثنا في المقدمة العامة كان لا بد من وجود تقنية تساعد السائقين على تحسين عملية السياقة وتجنب الحوادث ومنها السيارات ذاتية القيادة و العديد من الخدمات الاخرى التي تركز بشكل كبير على الرؤية الحاسوبية Computer Vision ..

لذلك يقوم هذا المشروع على بناء نظام تنبيه ذكي يستخدم الرؤية الحاسوبية لتقديم التنبيهات المناسبة في حالة نعاس السائق و في حالة اكتشاف لافتات مرورية وتصنيفها من اجل مساعدة السائق ويبني النظام على فكرتين اساسيتين تخدمان حالات الاستخدام المراد تطبيقها :

1- اكتشاف نعاس السائق من خلال اكتشاف الوجه اولاً Face detection ومن ثم تحديد نقاط معالم الوجه Facial landmarks points و استخدام خوارزمية معينة تعتمد على نسبة أبعاد العين Eye aspect ratio EAR و نسبة أبعاد الفم Mouth aspect ratio MAR ومن ثم تحديد عتبة Threshold معينة متفق عليها لاكتشاف نعاس السائق.

2- استخدام تقنية الكشف عن الكائنات Object Detection لاكتشاف اللافتات المرورية في الصورة (او الفيديو) وتصنيفها وتنبيه السائق بها ويختلف ال object detection عن ال classification بأنه يتم على مرحلتين :

1- اكتشاف مواقع الكائنات وتحديد لها

2- تصنيف الكائنات باستخدام الشبكات العصبونية الملتفة CNN

## الفصل الثاني: الدراسة المرجعية

قبل البدء بتصميم النظام ومعرفة متطلباته يجب القيام بدراسة مرجعية للمشروع المقترح وذلك من خلال البحث عن تطبيقات مشابهة ومعرفة الأساس الهندسي المبني عليه هذه التطبيقات ومنه يمكن استنباط المتطلبات والوظائف الأساسية ويمكن المقارنة بينهم محاولة لتحسين البناء التقني والهندسي للمشروع الذي نعمل عليه .

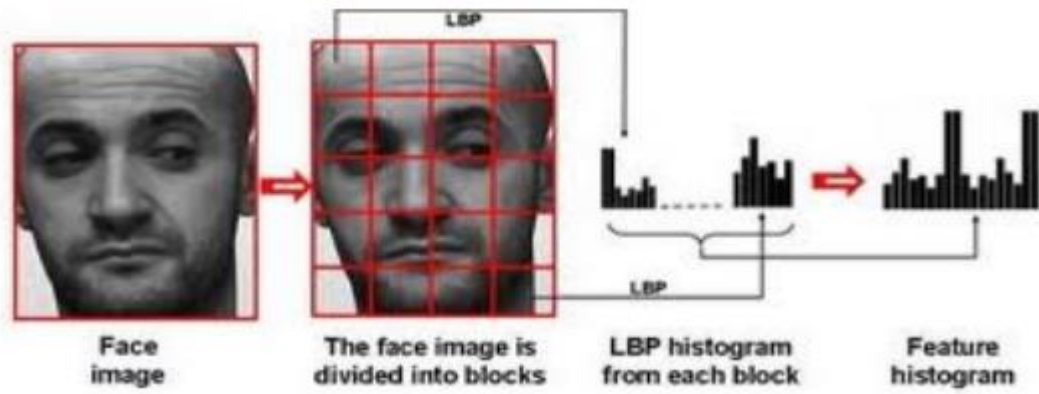
### 🌀 ظهور أنظمة كشف نوم السائقين:

لهدف من هذه الأنظمة هي الحد من الحوادث التي كان سببها القيادة أثناء النعاس, حيث حسب الإحصاءات فقد تبين أنه شيء مستهان به في وسط السائقين هي مشكلة حقيقية تواجه مالين السائقين حول العالم.

### تطبيقات مشابهة:

#### 1- Chandigarh University Driver Drowsiness Detection System :

ظهر هذا النظام في جامعة Gharuan Chandigarh في الهند من قبل الباحث Saini Vandna وهو نظام يعمل على جهاز خاص مربوط بكمرة معلقة تحت المرأة الأمامية .يعتمد النظام على LBP في تحديد Interests Of Range في ال Frame الملتقط ومن ثم تحديد ما إذا كانت العيون مغلقة أم لا .

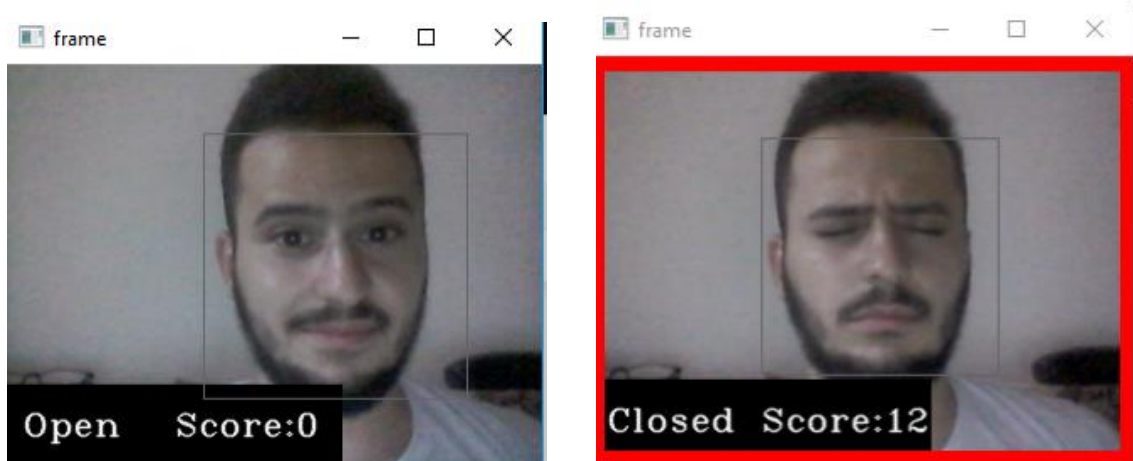


## 2- Damascus University's Driver Drowsiness Detection System :

ظهر هذا النظام في جامعة دمشق على يد الطلاب ( زكريا صافي , محمد حمزة شموط , هيام نجم الدين ) وهو نظام يعمل على جهاز خاص مربوط بكمرة معلقة تحت المرأة الأمامية أيضاً , يعتمد هذا النظام على

استخدام ال Haar Cascades

الذي يقوم باكتشاف الوجه والعينين Region of Interest ROI ومن ثم يتم تمرير جزء العين الايمن واليسر الى شبكة عصبونية ملتفة CNN والتي تحدد إن كان العين مغلقاً ام لا .





## مقارنة بين نظامنا ( قسم كشف النعاس ) والانظمة السابقة :

الميزات (Features)	First System	Second System	Our System
اكتشاف الوجه	يوجد	يوجد	يوجد
Real time	لا يوجد	يوجد	يوجد
التنبه الصوتي	يوجد	يوجد	يوجد
كشف التثاؤب	لا يوجد	لا يوجد	يوجد
سهولة الاستخدام	لا يوجد	لا يوجد	يوجد
Multiplatform	لا يوجد	لا يوجد	يوجد
FPS	2-5 FPS because of using LBP	5-15 FPS because of using Haar Cascades	25-30 FPS because of using Mediapipe And ML KIT

## 🌀 ظهور أنظمة اكتشاف اللافتات المرورية:

التعرف على العلامات المرورية هي تقنية تستطيع المركبة من خلالها التعرف على إشارات المرور الموضوعة على الطريق ، على سبيل المثال. "حد السرعة" أو "الأطفال" أو "المضي قدماً".

### 1- Traffic sign recognition based on deep learning system:

ظهر هذا النظام على يد الباحثين Yanzhao Zhu و Wei Qi Yan , ويعتمد هذا النظام على معمارية YOLOv5 التي تساعد على اكتشاف اللافتة المرورية وتصنيفها .



## 2- Traffic sign Classification:

يعتمد هذا النظام على استخدام الشبكات العصبية الملتفة CNN لتصنيف اللافتات المرورية من دون

اكتشافها او ايجاد مكانها في الصورة او الفيديو Localization .



مقارنة بين نظامنا ( قسم اكتشاف لافتات المرور) والانظمة السابقة :

الميزات ( Features )	First System	Second System	Our System
اكتشاف اللافتة المرورية	يوجد	لا يوجد	يوجد
تصنيف اللافتة المرورية	يوجد	يوجد	يوجد
تنبيه السائق كتابياً	لا يوجد	لا يوجد	يوجد
Bounding Box	يوجد	لا يوجد	يوجد
عدد اللافتات المرورية	8	30	20

## الفصل الثالث: دراسة المتطلبات

في هذا الفصل سنقوم بتحديد متطلبات الوظيفية والغير وظيفية للنظام ومن ثم سنحدد من سينتفع من هذه الوظائف Actors and stack Holders وتحديد حالات الاستخدام ونختتم هذا الفصل بتوصيف واجهة المستخدم .

### متطلبات النظام

#### قائمة المتطلبات الوظيفية

رقم المتطلب	شرح موجز عن المتطلب
1	كشف النعاس عن طريق اكتشاف اغلاق العينين لمدة معينة
2	كشف النعاس عن طريق اكتشاف التثاؤب
3	تنبيه السائق صوتياً عند اكتشاف نعاسه
4	اكتشاف اللافتات المروية اثناء القيادة وتصنيفها
5	تنبيه السائق كتابياً بتصنيف اللافتة المرورية التي رصدها

#### متطلبات أخرى للنظام(غير وظيفية)

رقم المتطلب	شرح موجز عن المتطلب
1	Scalability قابلية النظام للتوسع، حيث يمكن زيادة موديلات أخرى للكشف عن أحداث أخرى تتعلق بالطبيعة وزيادة عدد أصناف السلوك البشري الممكن تصنيفها وغيرها من الخدمات الأخرى.
2	Reliability موثوقية نتائج النظام ويتم تحقيق ذلك من خلال اختبار كل جزء على حدى والاختبارات النهائية للنظام
3	Response Time يعمل النظام ضمن زمن استجابة مقبول نسبياً ويتم تحقيق ذلك من خلال اختيار أدوات وخوارزميات سريعة وتعمل بزمن مقبول.
4	Accessibility سهولة الوصول للنظام واستخدامه بدون الحاجة الى Setup كامل في حالة تطبيق الهاتف.

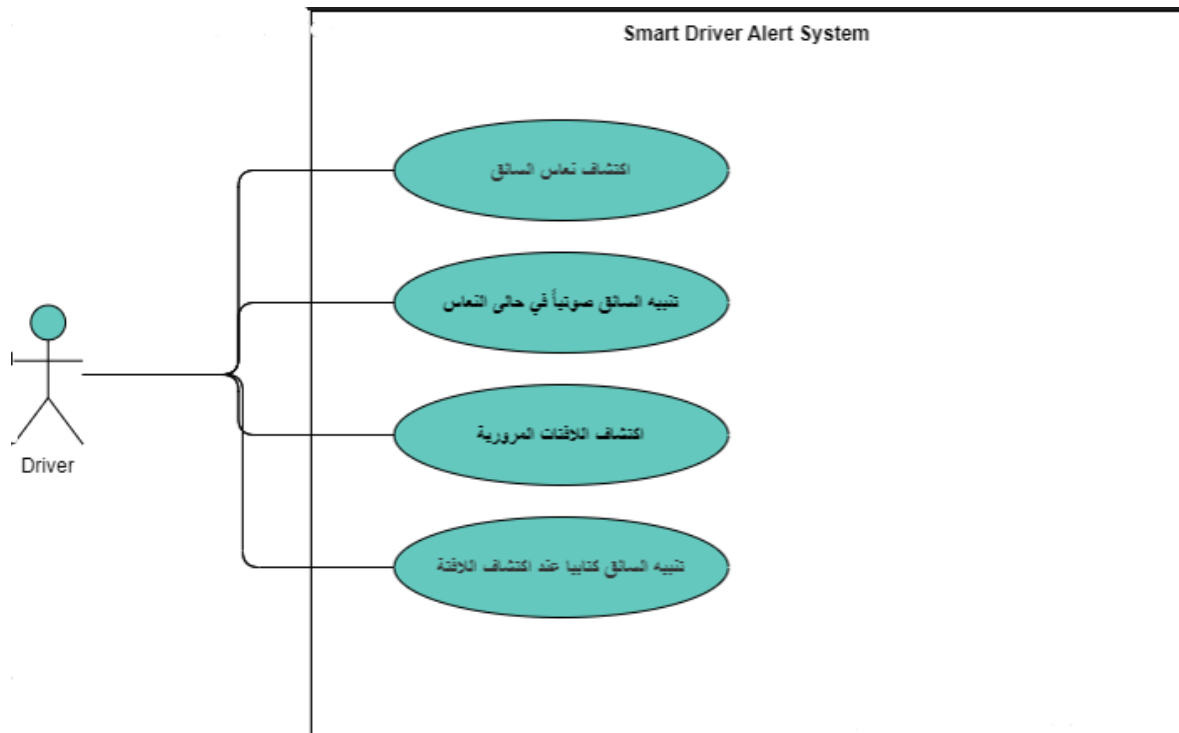
## توصيف المتطلبات الوظيفية

سنقوم في هذا القسم بتوصيف حالات الاستخدام حيث ان نظريا لطبيعته البسيطة، يعد مخطط حالة الاستخدام وسيلة جيدة للتواصل مع أصحاب الشأن. محاولة تمثيل العالم الحقيقي وتقديم عرض لأصحاب الشأن لفهم كيف سيتم عمل وتصميم النظام. قام "ساو" و"لي" بإجراء بحث حول فعالية مخططات حالة الاستخدام من عدمها، ووجدوا بأن مخططات حالة الاستخدام عبرت عن الهدف من النظام بشكل أكثر بساطة لأصحاب الشأن وأن مخطط الفئة قد أدى دورا مكملًا له في تفسير النظام. الهدف من مخطط حالة الاستخدام هو ببساطة تقديم رؤية مختصرة وعرض المتطلبات بشكل مناسب للأشخاص العاديين.

### أصحاب المنفعة

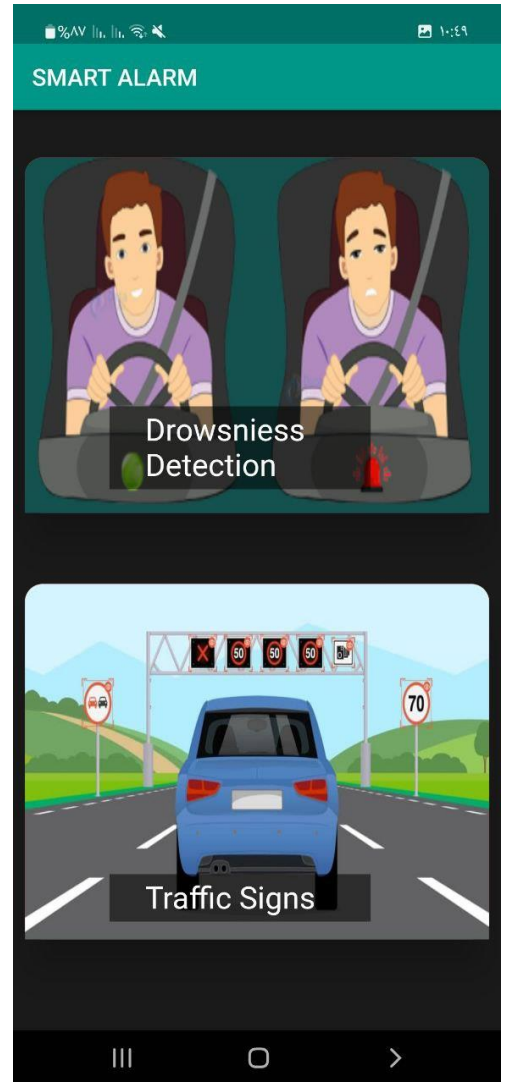
سائقين سيارات الشحن ووسائل النقل بين المحافظات و سائقي وسائل النقل العاملين لساعات طويلة .

### مخطط حالات الاستخدام



## توصيف واجهة الاستخدام

في هذا القسم سنقوم بتوصيف واجهة الاستخدام لحالات الاستخدام الجوهرية التي قمنا بتوصيفها في القسم السابق.



## الفصل الرابع: الموارد والتقنيات المستخدمة:

### 1- Python:

هي لغة برمجة، عالية المستوى سهلة التعلم مفتوحة المصدر قابلة للتوسيع، تعتمد أسلوب البرمجة الكائنية (OOP). لغة بايثون هي لغة مُفسّرة، ومُتعددة الاستخدامات، وتستخدم بشكل واسع في العديد من المجالات، كبناء البرامج المستقلة باستخدام الواجهات الرسومية وفي تطبيقات الويب، ويمكن استخدامها كلغة برمجة نصية للتحكم في أداء العديد من البرمجيات مثل بلندر. بشكل عام، يمكن استخدام بايثون لعمل البرامج البسيطة للمبتدئين، ولإنجاز المشاريع الضخمة في الوقت نفسه. غالباً ما يُنصح المبتدؤون في ميدان البرمجة بتعلم هذه اللغة لأنها من بين أسرع اللغات البرمجية تعلماً. غالباً ما تستخدم في الدراسات الأكاديمية للذكاء الاصطناعي بسبب الإنتاجية المتزايدة التي توفرها. ن تتميز بايثون بمجتمعها النشط، كما أن لها الكثير من المكتبات البرمجية ذات الأغراض الخاصة التي برمجها أشخاص من ذلك المجتمع. مثلاً، هناك مكتبة باي جايم التي توفر مجموعة من الدوال من أجل برمجة الألعاب. يمكن لبايثون أيضاً التعامل مع العديد من أنواع قواعد البيانات مثل ماي إس كيو إل وغير ذلك. تدعم بايثون أنماط برمجية متعددة هي التوجيه الكائني، البرمجية جانبية التوجيه والبرمجة الوظيفية. تُستخدم بايثون عادةً مثل العديد من لغات البرمجة الديناميكية كلغة برمجة نصية. بايثون لديها نموذج مفتوح للتطوير، قائم على مجتمع بايثون البرمجي ومدعوم من مؤسسة برمجيات بايثون. والتي تحافظ على تعريف اللغة في التنفيذ المرجعي لسي بايثون.

### 2- Flutter:

هي برنامج مفتوح المصدر لمجموعة أدوات تطوير البرمجيات تم تطويرها من قبل جوجل لبناء واجهات المستخدم لتطبيقات نظام الأندرويد ونظام أي أو إس ونظام الويندوز وتطبيقات الويب. تم إنشاء اطار عمل فلاتر فلاتر من الصفر واستخدم لكتابته وبنائه لغة دارت ولغة سي++ وهو لا يزال نوعاً ما في مرحلتها التجريبية، على الرغم من ذلك، فاطر عمل فلاتر تم اصداره رسمياً لكن لا يزال في المرحلة التجريبية للنجاح، يستخدم اطار عمل فلاتر في الأساس لتطوير واجهات الاستخدام ويتعاون مع لغة البرمجة دارت للتعامل مع العمليات البرمجية جهة البرنامج الخلفية.

تم تحسين وتطوير اطار عمل فلاتر لجعله أداة عملية قوية لتحسين سرعة وكفاءة التطبيقات، كما تعتبر هذه نعمة للمؤسسات والمطورين ومن يسعى للعمل على المنصتين بأقل وأوفر جهد وتكلفة.

يساعد اطار عمل فلاتر المصممين والمبرمجين على تشييد تطبيقات بتصميم عصري وجذاب من خلال حزم ومكتبات مبنية ومكتوبة مسبقاً وجاهزة للاستخدام مما سيسهل عليك عملك كثيراً.

### 3- TensorFlow:

هي مكتبة برمجية مجانية ومفتوحة المصدر في مجال تعلم الآلة مقدمة من شركة غوغل . تستخدم في العديد من المجالات الفرعية ولكن لها تركيز محدد في تدريب واستدلال الشبكات العصبية العميقة.

- Tensorflow light:

وهي عبارة عن مجموعة من الأدوات التي تتيح التعلم الآلي على الجهاز من خلال مساعدة المطورين على تشغيل نماذجهم على الأجهزة المحمولة والمدمجة والمتطورة.

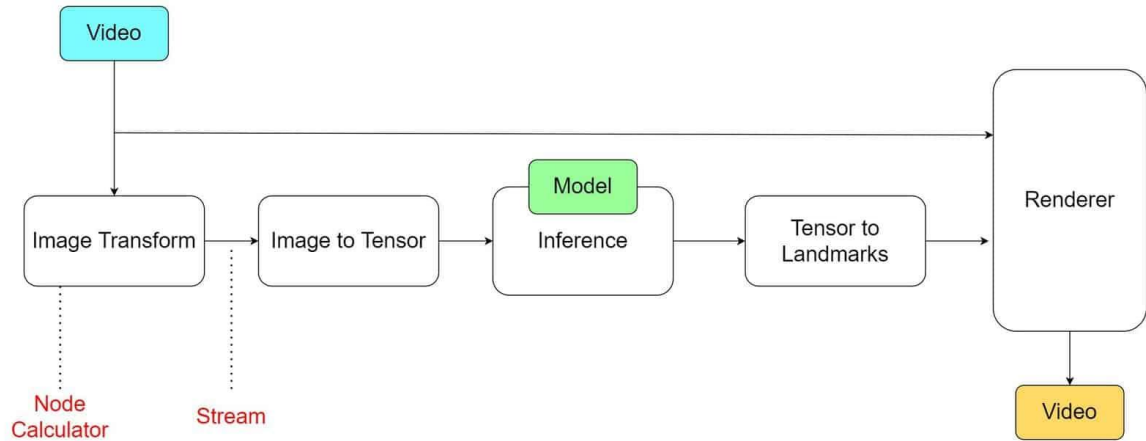
- Tensorflow Object detection API:

في الماضي ، كان إنشاء Custom Object Detection Model يبدو وكأنه مهمة صعبة وتستغرق وقتاً طويلاً. الآن ، باستخدام أدوات مثل TensorFlow Object Detection API ، يمكننا إنشاء نماذج موثوقة بسرعة وسهولة. تسمح هذه الـ Api بتوليد Object Detection Model عن طريق تجميع الـ Data set و القيام بعملية الـ Labeling ومن ثم اختيار موديل مدرب مسبقاً Pretrained model مناسب ومن ثم ستساعدك على بناء الموديل المخصص وبدقة وسرعة عاليتين.

### 4- MediaPipe:

MediaPipe هو إطار عمل لبناء خطوط أنابيب تعلم الآلة لمعالجة بيانات السلاسل الزمنية مثل الفيديو والصوت وما إلى ذلك. تسهل MediaPipe أيضًا نشر تقنية التعلم الآلي في العروض التوضيحية والتطبيقات على مجموعة متنوعة من منصات الأجهزة المختلفة (مثل Android و iOS والأجهزة المضمنة مثل Raspberry Pi)

تدعم MediaPipe المنتجات والخدمات الثورية التي نستخدمها يوميًا. على عكس أطر التعلم الآلي المتعطشة للطاقة ، يتطلب MediaPipe الحد الأدنى من الموارد. إنه صغير جدًا وفعال لدرجة أنه حتى أجهزة إنترنت الأشياء المدمجة يمكنها تشغيله. في عام 2019 ، فتحت MediaPipe عالمًا جديدًا بالكامل من الفرص للباحثين والمطورين بعد إصدارها العام.



تقدم MediaPipe العديد من الخدمات يمكن الاطلاع عليها من الروابط التالية :

1. [Face Detection](#)
2. [Face Mesh](#)
3. [Iris](#)
4. [Hands](#)
5. [Pose](#)
6. [Selfie Segmentation](#)
7. [Object Detection](#)

#### 4- ML Kit:

ML Kit هي مجموعة SDK للجوال تمكّنك من إضافة ميزات تعلم الآلة القوية إلى تطبيق الهاتف المحمول. وهو يدعم كلاً من Android و iOS ويقدم نفس الميزات لكلا النظامين الأساسيين. تعد SDK جزءاً من Firebase وتجمع معاً العديد من تقنيات التعلم الآلي من Google مثل Cloud Vision API و TensorFlow Lite. يأتي SDK مع مجموعة من واجهات برمجة التطبيقات الجاهزة للاستخدام لحالات استخدام الأجهزة المحمولة الشائعة مثل اكتشاف الوجه والتعرف على النص ومسح الرمز Barcode ووضع العلامات على الصور والتعرف على معالم الوجه. يتم تقديمها إما على أنها واجهات برمجة تطبيقات على الجهاز أو على السحابة. تتمتع واجهات برمجة التطبيقات على الجهاز بميزة القدرة على معالجة البيانات بسرعة ، فهي مجانية الاستخدام ولا تتطلب اتصالاً بالشبكة للعمل. توفر واجهات برمجة التطبيقات المستندة إلى السحابة مستوى أعلى من الدقة لأنها قادرة على الاستفادة من قوة تقنيات التعلم الآلي في Google Cloud Platform. جميع واجهات برمجة التطبيقات المستندة إلى السحابة هي خدمات متميزة ، مع حصة مجانية في المكان.



## الفصل الخامس: اكتشاف النعاس

### النوم خلال القيادة :



العالم على دراية بمخاطر الشرب والقيادة أو حتى إرسال الرسائل النصية والقيادة ، لكن الكثير من الناس يقللون من مخاطر القيادة بالنعاس. في كل عام ، تتسبب القيادة التي تعاني من النعاس في حدوث حوالي 2,100,000 حادث و 1,149,000 إصابة و 32,550 حالة وفاة في العالم ، وفقاً للرابطة الإحصائية الأمريكية American Association Statistical. تساهم القيادة النعاسية في حدوث ما يقدر بـ 5.9 ٪ من جميع الحوادث ، وفقاً لـ ASA. ومع ذلك ، قد يكون الرقم الفعلي أعلى من ذلك بكثير لأنه من الصعب تحديد ما إذا كان السائق يشعر بالنعاس في وقت وقوع الحادث. في كثير من الحالات ، تكون القيادة بالنعاس خطرة مثل القيادة أثناء إعاقتها للكحول.

كم مرة يقود الناس وهم نعاس؟

- حوالي 27 ٪ من السائقين عن القيادة وهم متعبون جداً لدرجة أنهم يجدون صعوبة في إبقاء أعينهم مفتوحة (ASA).

- 1 من كل 25 سائق يعترف بالنوم خلف عجلة القيادة (CDC) .

- 47 ٪ ممن نزلوا برأسهم أثناء القيادة فعلوا ذلك في رحلة استغرقت ساعة واحدة أو أقل (NHTSA)

متى يحدث النعاس أثناء القيادة ؟

تحدث معظم حوادث القيادة التي تسبب النعاس بين منتصف الليل والسادسة صباحاً أو في وقت الحق بعد الظهر عندما ينظم الجسم النوم غالباً ما تتضمن حوادث القيادة النعاسية فقط راكباً واحداً يخرج خارج الطريق تعد حوادث القيادة أثناء النعاس أكثر شيوعاً على الطرق السريعة والطرق الريفية (NHTSA)

ما هي القيادة أثناء النعاس؟

القيادة أثناء النعاس ، والمعروفة أيضاً بإرهاق السائق أو القيادة المتعبة ، هي فعل القيادة أو تشغيل مركبة بمحرك أثناء التعب والشعور بالإرهاق أو النعاس.

ضغوط العمل أو الليالي المتقطعة مع الأطفال الصغار من الأسباب الشائعة لقلة النوم. ومع ذلك ، يمكن أن تساهم عوامل أخرى في القيادة بالنعاس ، مثل الأدوية التي تتناولها أو اضطراب النوم غير المعالج الذي يتركك مستنزفاً وغير قادر على البقاء مستيقظاً أثناء النهار.

يتأثر العاملون في وقت متأخر من الليل والثالث بشكل خاص بالإفراز الطبيعي للميلاتونين المرتبط بالساعات المظلمة أثناء عودتهم إلى المنزل بعد نوبة طويلة.

تتمثل الآثار الرئيسية للقيادة بالنعاس في عدم القدرة على التركيز ، وتأخر رد الفعل ، وسوء الحكم ، وعدم القدرة على الحكم على المسافات والسرعات ، وبالطبع النوم.

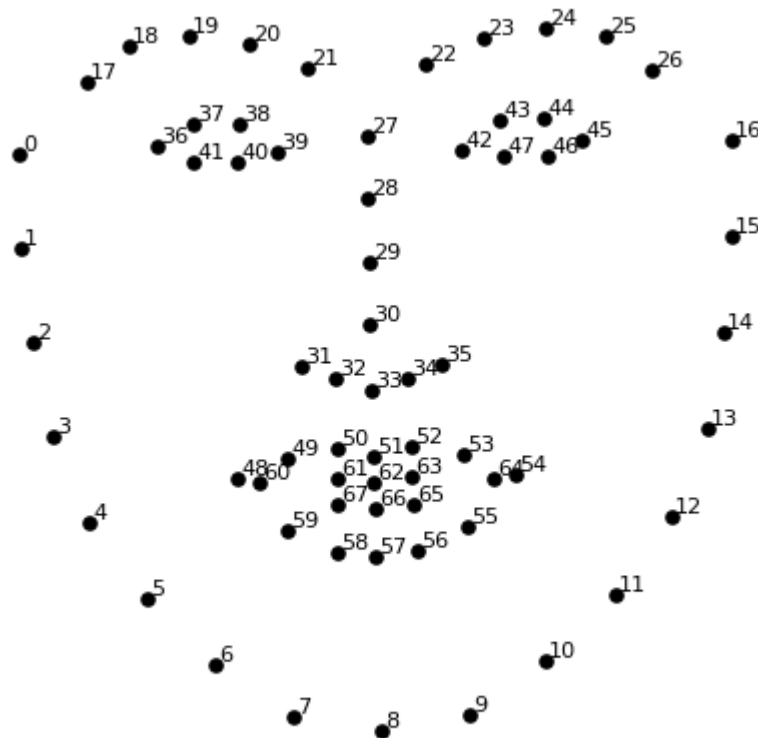
آثار القيادة بالنعاس أكثر حدة مما يدركه معظم الناس. عندما تكون مستيقظاً لأكثر من 18 ساعة ، يكون التأثير على جسمك هو نفسه كما لو كان لديك BAC بنسبة 0.05 في المائة. وفقاً لمركز السيطرة على الأمراض بعد 24 ساعة من الاستيقاظ ، فإن الأمر يشبه الحصول على BAC بنسبة 0.10 في المائة ، وهو ما يتجاوز بكثير الحد القانوني في أغلب الدول . بالنظر إلى أن الحد القانوني لمحتوى الكحول في الدم هو 0.08 بالمائة ، فإن القيادة بالنعاس تشبه القيادة تحت تأثير الكحول

ما مدى شيوع القيادة بالنعاس؟ كشفت دراسة CDC لما يقرب من 150.000 سائق بالغ في 19 ولاية وواشنطن العاصمة عن بعض الأفكار المثيرة للاهتمام. اعترف حوالي 4 بالمائة من البالغين بأنهم ناموا أثناء القيادة خلال الثلاثين يوماً الماضية. وفي الوقت نفسه ، كان السائقون الذين ينامون أقل من ست ساعات كل ليلة أو يتعرضون للشخير أكثر عرضة للنوم أثناء القيادة. يقدر أن واحداً من كل 25 سائقاً يبلغون من العمر 18 عاماً أو أكثر قد ناموا خلال الثلاثين يوماً الماضية.

## اكتشاف معالم الوجه : Face Landmarks Detection

يعد اكتشاف معالم الوجه مهمة الرؤية الحاسوبية حيث نريد اكتشاف النقاط الرئيسية وتتبعها من وجه بشري. هذه المهمة تنطبق على العديد من المشاكل.

على سبيل المثال ، يمكننا استخدام النقاط الأساسية لاكتشاف موضع رأس الإنسان ودورانه. من خلال ذلك ، يمكننا تتبع ما إذا كان السائق ينتبه أم لا. أيضًا ، يمكننا استخدام النقاط الأساسية لتطبيق الواقع المعزز بشكل أسهل. وهناك العديد من الحلول التي يمكننا إنشاؤها بناءً على هذه المهمة.

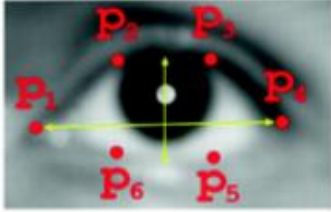


The image is created by Brandon Amos from CMU that creates OpenFace

وسيتم استخدام هذه التقنية في هذا النظام لحساب ال EAR وال MAR واللذان سيساعدان باكتشاف نعاس السائق ولكن ما هو ال EAR وال MAR ؟

### Eye Aspect Ratio (EAR)

EAR ، كما يوحي الاسم ، هي نسبة طول العين إلى عرض العينين . يتم حساب طول العيون من خلال حساب متوسط أكثر من خطين عموديين مختلفين عبر العينين كما هو موضح في الشكل أدناه.

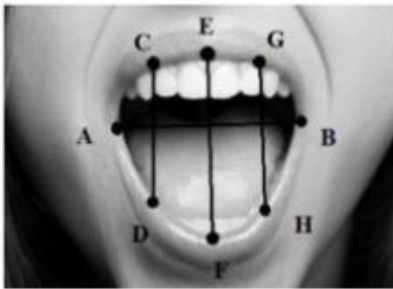


$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

كانت فرضيتنا أنه عندما يشعر الفرد بالنعاس ، فمن المرجح أن تصبح عيونه أصغر ومن المرجح أن ترمش أكثر. بناءً على هذه الفرضية ، يمكن التنبؤ بنعاس السائق إذا أغلق السائق عيونه لمدة معينة فإنه غالباً يغفو قليلاً فيجب ان يتم انذاره .

### Mouth Aspect Ratio (MAR)

على غرار EAR من الناحية الحسابية ، فإن MAR ، كما تتوقع ، يقيس نسبة طول الفم إلى عرض الفم . كانت فرضيتنا أنه عندما يصاب الفرد بالنعاس ، فمن المرجح أن يتثائب ويفقد السيطرة على أفواههم ، مما يجعل MAR الخاص بهم أعلى من المعتاد في هذه الحالة.



$$MAR = \frac{|EF|}{|AB|}$$

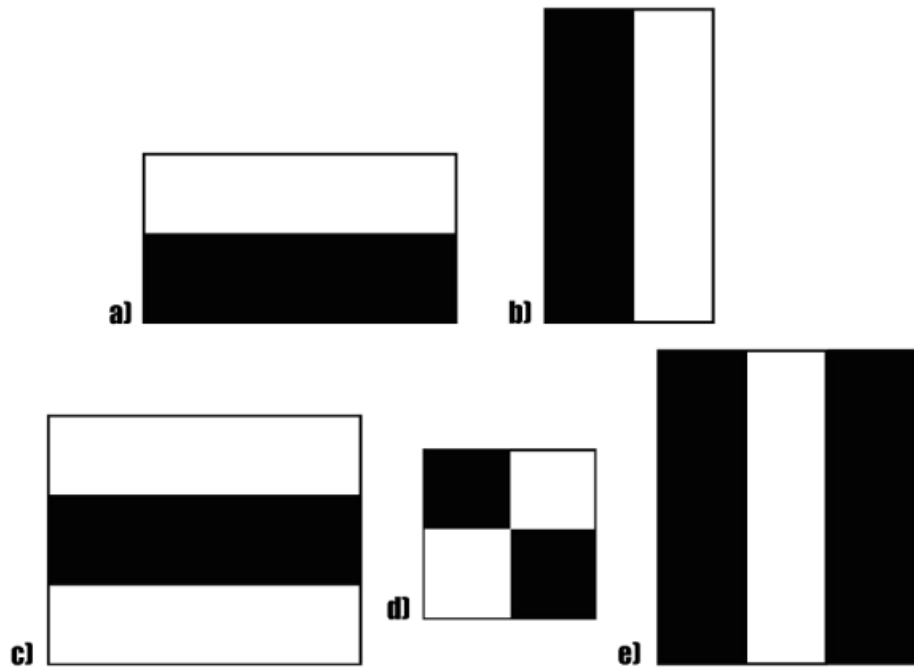
قبل اختيار الطريقة والتقنية المناسبة لتنفيذ فكرة اكتشاف النعاس كان علينا اجراء التجارب على العديد من التقنيات المشهورة ومقارنتها من حيث السرعة والدقة بما يفيد النظام المقترح باكبر شكل ممكن . مقارنة شاملة بين تقنيات اكتشاف الوجه ومعالمه :

## Haar Cascade Classifier:

هي خوارزمية الكشف الكائنات تستخدم لتحديد الوجوه في صورة أو فيديو في الوقت الفعلي. تستخدم الخوارزمية ميزات الكشف عن الحواف أو الخطوط التي اقترحتها فيوال وجونز في ورقتهما البحثية "الكشف السريع عن الأشياء باستخدام سلسلة معززة من الميزات البسيطة" المنشورة في عام 2001. يتم إعطاء الخوارزمية الكثير من الصور الإيجابية التي تتكون من الوجوه ، والكثير من صور سلبية ال تتكون من أي وجه للتدريب عليه.

يختار Adaboost بشكل أساسي أفضل الميزات ويدرب المصنفين على استخدامها. تستخدم مجموعة من "المصنفات الضعيفة" لإنشاء "مصنف قوي" يمكن للخوارزمية استخدامه لاكتشاف الكائنات. يتم إنشاء المتعلمين الضعفاء عن طريق تحريك نافذة فوق صورة الإدخال ، وحساب ميزات Haar لكل قسم فرعي من الصورة. تتم مقارنة هذا الاختلاف بالحد الذي تم تعلمه والذي يفصل بين الكائنات غير الكائنات. لأن هذه "مصنفات ضعيفة" ، هناك حاجة لعدد كبير من ميزات Haar للدقة لتشكيل مصنف قوي.

إدخال ميزات haar الموضحة أدناه كانت هي المساهمة الأولى في البحث. تسهل هذه الميزات الموجودة في الصورة اكتشاف الحواف أو الخطوط في الصورة ، أو اختيار المناطق التي يوجد بها تغيير مفاجئ في شدة وحدات البكسل.



### Dlib (HOG + SVM):

يعتمد كاشف الوجه هذا على HOG (Histogram of Oriented Gradients) و SVM (Support Vector Machine) وهو أكثر دقة بشكل ملحوظ من سابقتها. التقنية المستخدمة في هذه التقنية ليست ثابتة على التغيرات في زاوية الوجه ، لذلك فهي تستخدم خمسة مرشحات HOG مختلفة:

1. Frontal face
2. Right side turned face
3. Left side turned face
4. Frontal face but rotated right
5. Frontal face but rotated left

A HOG هو واصف ميزة يستخدم بشكل عام لاكتشاف الكائنات. تشتهر HOGs على نطاق واسع باستخدامها في اكتشاف المشاة. يعتمد HOG على خاصية الكائنات داخل الصورة لامتلاك توزيع تدرجات الكثافة أو اتجاهات الحافة. يتم حساب التدرجات داخل صورة لكل كتلة. تعتبر الكتلة بمثابة شبكة بكسل تتكون فيها التدرجات من حجم واتجاه التغير في شدة البكسل داخل الكتلة.

### Dlib(HOG + MMOD):

يعتمد هذا الكاشف على نهج التعلم العميق (شبكة الالتفاف العصبية) ويستخدم طريقة Maximum-Margin Object Detection (MMOD) لاكتشاف الوجوه في الصور. تم تدريب هذا أيضًا على حد أدنى لحجم الوجه 80 × 80 ويوفر خيار اختزال الصور. هذا الكاشف بطيء جدًا على وحدة المعالجة المركزية ولكن يمكن استخدامه على وحدة معالجة الرسومات NVIDIA ويتفوق على أجهزة الكشف الأخرى في السرعة على وحدة معالجة الرسومات.

### OpenCV Deep learning based face detection:

يعتمد هذا الكاشف على نهج التعلم العميق ويستخدم ResNet-10 Architecture لاكتشاف وجوه متعددة في مرور واحد (Single Shot Detector SSD) للصورة عبر الشبكة (نموذج). تم تضمينه في OpenCV منذ أغسطس 2017 ، مع الإصدار الرسمي للإصدار 3.3 ، ومع ذلك ، فهو ليس شائعًا مثل OpenCV Haar Cascade Face Detector ولكنه بالتأكيد أكثر دقة ويستخدم غالبًا Caffee face detection model.

Mediapipe :

تم شرح هذا الكاشف سابقاً ولكن يعتمد هذا الكاشف أيضاً على نهج التعلم العميق ويستخدم BlazeFace وهو كاشف للوجه خفيف الوزن للغاية ودقيق للغاية ومستوحى ومعدل من Single Shot MultiBox (SSD) Detector و MobileNetv2 وبما أنه light weight model فهو اختيار ممتاز للأجهزة الضعيفة مثل الهواتف و الأنظمة المضمنة و ال Raspberry PI .

ML KIT Face detection :

باستخدام واجهة برمجة تطبيقات اكتشاف الوجه في ML Kit ، يمكنك اكتشاف الوجوه في صورة ما وتحديد ميزات الوجه الرئيسية والحصول على ملامح الوجوه المكتشفة. لاحظ أن واجهة برمجة التطبيقات تكتشف الوجوه ، ولا تتعرف على الأشخاص. من خلال اكتشاف الوجه ، يمكنك الحصول على المعلومات التي تحتاجها لأداء مهام مثل تجميل صور السيلفي والصور الشخصية ، أو إنشاء صور رمزية من صورة المستخدم. نظرًا لأن ML Kit يمكنها إجراء اكتشاف الوجه في الوقت الفعلي ، يمكنك استخدامها في تطبيقات مثل دردشة الفيديو أو الألعاب التي تستجيب لتعبيرات اللاعب.

ML kit	Mediapipe	Opencv dnn	Dlib dnn	HOG	Haar	Features
25-fps	25-60 fps	25-30 fps	0-3 fps	8 Fps	15 fps	FPS
yes	yes	slightly	yes	slightly	no	Face orientation
yes	yes	no	no	no	no	Face mesh
best	best	good	Bad on CPU	good	Bad	Accuracy
Best speed accuracy trade off and good for mobile and embedded devices	Best speed accuracy trade off and good for mobile and embedded devices	Accurate and fast on a CPU	More accurate and fast on GPU	Detects face on different orientation	Quick and easy to setup	Use Case

بعد المقارنة بين التقنيات والطرق التي يمكن فيها اكتشاف الوجه ومعالم الوجه وصلنا للنتائج التالية:

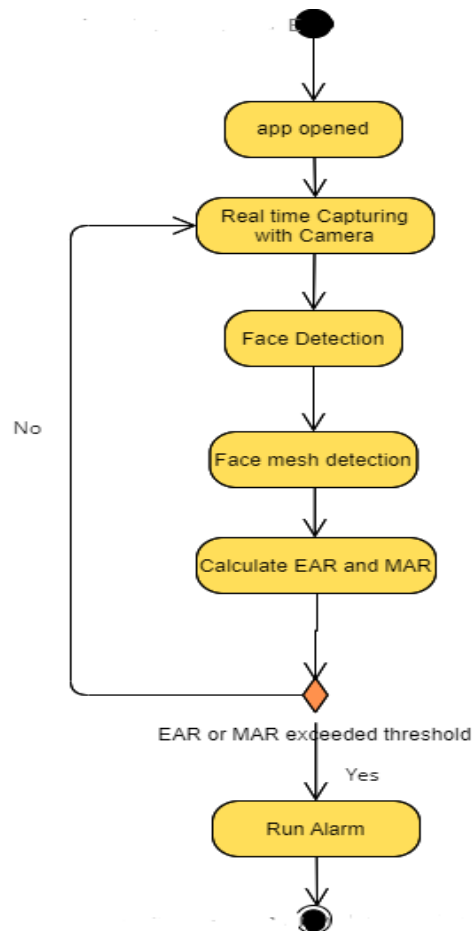
1. استخدام ال Haar cascade face detection سيعطي نتائج متوسطة من ناحية السرعة والدقة حيث تشتهر ال haar cascade باعطاء false positive بمعدل كبير كما ان ال fps لاكتشاف الوجه وصل الى 13 FPS وكانت الخطوة التالية هي اكتشاف العينين والفم لكن باضافة الميزيتين حصل frame drop out to 8fps اما الخطوة الاخيرة وهي تمرير جزء العينين وجزء الفم الى شبكة CNN ومن ثم تحديد النعاس أبطأ النظام بشكل اكبر ووصل إلى 5FPS ولذلك قررنا إيجاد بديل مناسب اكثر.
2. انتقلنا بعدها لاستخدام ال opencv face detection with deep learning فكانت النتائج جيدة نوعاً ما ولكن لا تقدم هذه التقنية ميزة اكتشاف العينين و الفم لذلك لم نخدم ما كنا نبحث عنه.
3. انتقلنا بعدها لفكرة حساب ال EAR و ال MAR والتي ستمنح سرعة اكبر لاننا لن نمرر النتائج عبر شبكة عصبونية فبذلك قد اختصرنا خطوة حسابية مهمة وتم تجربة ال Dlib HOG with SVM والتي اعطت نتائج جيدة في اكتشاف الوجه ولكن بطيئة خاصة اذا تم تفعيل خاصية ال landmark detection لإيجاد معالم الوجه ووصلنا الى 6 FPS ومن ثم جربنا Dlib with deep learning الذي كانت دقته كبيرة ولكن بطيء جداً لانه مبني ليتم استخدامه على ال GPU وليس ال CPU ونحن نبحث عن حل يناسب ال CPU والاجهزة الضعيفة مثل الهواتف وال raspberry pi لذلك قررنا ان نكمل البحث عن حل افضل.
4. انتقلنا بعدها الى تجربة ML KIT SDK والتي تقدم العديد من الخدمات منها اكتشاف الوجه ومعالمه ووصلنا الى نتائج جيدة جداً من حيث الدقة والسرعة حيث وصلت السرعة الى 25-30 fps وذلك على الهاتف المحمول لان هذه التقنية مبنية للعمل على اجهزة اندرويد و IOS لذلك قمنا باختيار هذه التقنية في تنفيذ النظام على تطبيق الهاتف المبني من خلال Flutter .
5. بما ان ال ML kit مبني فقط للتعامل مع الهواتف , اردنا البحث عن حل اخر يناسب الحواسيب والانظمة المضمنة مثل ال raspberry pi مما سيساعد على تنفيذ النظام على بيئات مختلفة فوجدنا تقنية رائعة وهي ال mediapipe التي تقدم أيضاً العديد من الخدمات منها اكتشاف الوجه و ال face mesh وتقدم أفضل tradeoff بين السرعة والدقة وبسرعة تصل الى 25-60 FPS لذلك قمنا باختياره وبنائه عبر لغة البرمجة python



## خطوات عمل النظام في كشف النعاس :

1. يأخذ النظام صورة من كاميرا الهاتف او كاميرا خارجية كمدخل ( الكاميرا تقوم بأخذ كل الحركات التي يقوم بها السائق أو ما يسمى الوقت الحقيقي لذلك يقوم النظام بتجزئة المشهد إلى إطارات ومعالجة كل إطار لوحده).
2. سيتم اكتشاف الوجه وتحديد المكان منطقة الاهتمام في ال frame
3. يتم ايجاد ال face landmarks او ال face mesh من منطقة الوجه وإيجاد احداثيات كل منهم (تم اختيار النقاط المتعلقة بالعينين والفم فقط لتقليل العمليات الحسابية).
4. حساب ال EAR وال MAR لكل frame
5. اذا تخطى ال EAR or MAR العتبة المسموحة لهم سيجد النظام ان السائق في حالة نعاس وسيتم تشغيل المنبه
6. وإلا عد للخطوة الاولى .

ويشرح ال work flow diagram التالي الخطوات بشكل أفضل :

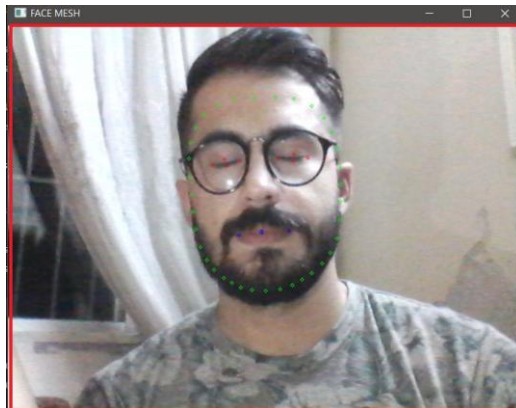


## التنفيذ: (Desktop and Mediapipe)

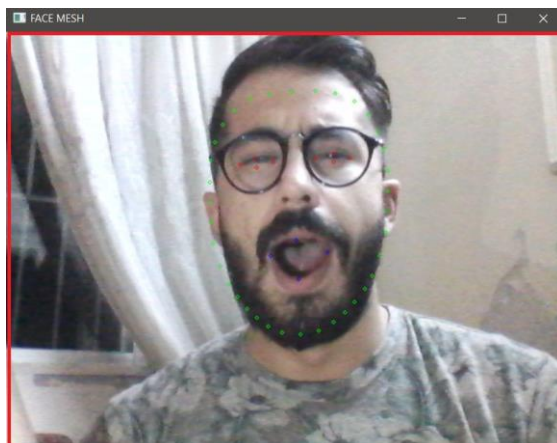
1- اكتشاف الوجه وال face landmarks :



2- اكتشاف اغلاق العين :



3- اكتشاف التثاؤب :



## التنفيذ: (Flutter and ML Kit)



Closed eyes detection



Face and landmark detection



Yawn detection

## الفصل السادس: اكتشاف اللافتات المرورية



### كشف اشارات المرور والتعرف عليها:

مع التطور السريع للتكنولوجيا أصبحت السيارات وسيلة نقل لا غنى عنها في السفر اليومي للأشخاص وعلى الرغم من أشعبيتها قد قدمت راحة كبيرة للناس إلا أنها تسببت أيضاً في العديد من مشاكل السلامة المرورية التي لا يمكن تجاهلها مثل الازدحام المروري وحوادث الطرق المتكررة. تنجم مشكلات السلامة المرورية إلى حد كبير إلى أسباب ذاتية تتعلق بالسائق مثل عدم الانتباه للإشارات أو عدم الامتثال لقواعد السير ونتيجة لذلك كان لابد من إيجاد نظام يساعد السائقين في كشف الإشارات المرورية والتعرف عليها خصوصاً عندما يكون السائق متعباً أو غير قادر على إعطاء الاهتمام المناسب للإشارات فتساعده على الانتباه لحدود السرعة وتبعده عن التستت أو الارتباط لأن الإشارات ممكن أن تتغير من قسم لآخر على نفس الطريق.

### تحديد الإشارات وتصنيفها:

بداية يجب التمييز بين 3 مفاهيم استخدمناها في مشروعنا ألا وهي التصنيف classification والموقع localization والكشف detection:

التصنيف: تصنيف الكائن بناءً على فئات أو أنواع محددة مسبقاً وتدخل فيه عدة خصائص مثل الموقع والشكل وغيره. يتم إعطاء الصورة ويكون المطلوب تصنيفها إلى فئة معينة مثل برتقالة أو تفاحة أو قطعة وما إلى ذلك فهو يجاوب على السؤال "ماذا يوجد في الصورة؟".

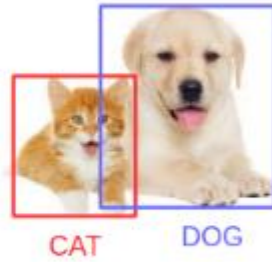
تحديد الموقع: هو تحديد موقع كائن واحد أو أكثر في الصورة عبر رسم مربع حول نطاقه.

الكشف: يجمع بين المفهومين السابقين فهو يمثل عملية تحديد الهوية ويقوم بتحديد إحداثيات وموقع كل كائن ويميز بين الكائنات المختلفة في صورة واحدة حيث يتم إعطاء صورة ويكون المطلوب اكتشاف وتوقع ما إذا كانت الفئات (برتقالة – تفاحة – قطة ...) موجودة في الصورة فهو يجاوب على السؤال هل تم اكتشاف الكائن (نعم / لا) فإذا تم اكتشافه يرسم مربع محيط حول كل كائن ويصنفه إلى فئته الخاصة .

Detection = classification + localization



Image Localization



Object Detection

ولأننا نريد معرفة نوع الإشارة وتحديد موقعها في المشهد استخدمنا مفهوم الكشف Detection .

أجهزة كشف الأغراض Objects Detection هي نوعين : ذات المرحلة الواحدة وذات المرحلتين وكلتاهما تقومان بالمهمتين التاليتين:

- البحث عن الكائنات في الصورة
- تصنيف كل كائن على حدا وإحاطة نطاقه بمربع

في الأجهزة ذات المرحلتين: يتم اقتراح مناطق الكائن التقريبية باستخدام ميزات عميقة قبل استخدامها للتصنيف بالإضافة إلى حساب انحدار الصندوق المحيط بالكائن وهذه النوع من الخوارزميات تحقق دقة عالية ولكنها بطيئة جداً ومن أمثلتها:

RCNN and SPPNet – Fast RCNN and Faster RCNN – Mask RCNN – Pyramid

Networks- GRCNN

أما في الأجهزة ذات المرحلة الواحدة: يتم التنبؤ بالمربعات المحيطة بالكائن بدون خطوة اقتراح المنطقة لذلك تستغرق وقتاً أقل وبالتالي يمكن استخدامها في تطبيقات الزمن الحقيقي real time وهي سريعة جداً وأبسط من الناحية الهيكلية بالنسبة لأجهزة ذات المرحلتين ومن أمثلتها:

YOLO – SSD – Retina Net

## التقنيات المستخدمة:

استخدمنا المودل SSD MobileNet v2 FPNLite 320x320 px وهو مودل تم إنشاؤه باستخدام Tensorflow object detection API ويدعم الصور الملونة RGB وبإمكانه تحديد أكثر من 10 كائنات في صورة واحدة وهو قائم على tensorflow Lite الخيف الوزن للأجهزة المحمولة والمدمجة فيتيح استدلال التعلم الآلي على الجهاز بزمان انتقال منخفض وحجم صغير بأدنى حد وهذا المودل موزد بميزة فقدان التركيز focal loss الخاصة بالهواتف (هي نسخة Retinanet in Lin et al مخصصة للهواتف تسمح بالتدريب على مجموعة قليلة من الأمثلة الصعبة بحيث تمنع الكشف من الارتباك أثناء التدريب وتكون شبكة Retinanet قادرة على مطابقة سرعة أجهزة الكشف ذات المرحلة الواحدة مع تجاوز دقة الأجهزة ذات المرحلتين).

سبب استخدام هذا المودل: نظراً لأن مشروعنا يخص تطبيقات الهواتف الذكية لذلك كان لابد من البحث عن مودل لكشف الأغراض بحيث نحقق موازنة بين السرعة والدقة ( سريع في الزمن الحقيقي ودقته ليست سيئة ) وهو يقدم ميزات أخرى مقارنة بغيره فهو يقوم بضغط الصور المدخلة لتصبح بقياس 320x320 وبذلك بهدف التسريع ثم يطبق عمليات إضافية على الصور الدخلة data augmentation مثل زيادة إضاءتها أو قلبها flip أو إزاحتها shift وذلك بهدف زيادة حجم مجموعة البيانات Dataset للحصول على دقة أفضل وبعد ذلك يفك ضغط الصورة ويعرضها بحجمها الطبيعي الذي أدخلت به وسنذكر تفاصيل كل هذه الخطوات بشكل مفصل لاحقاً.

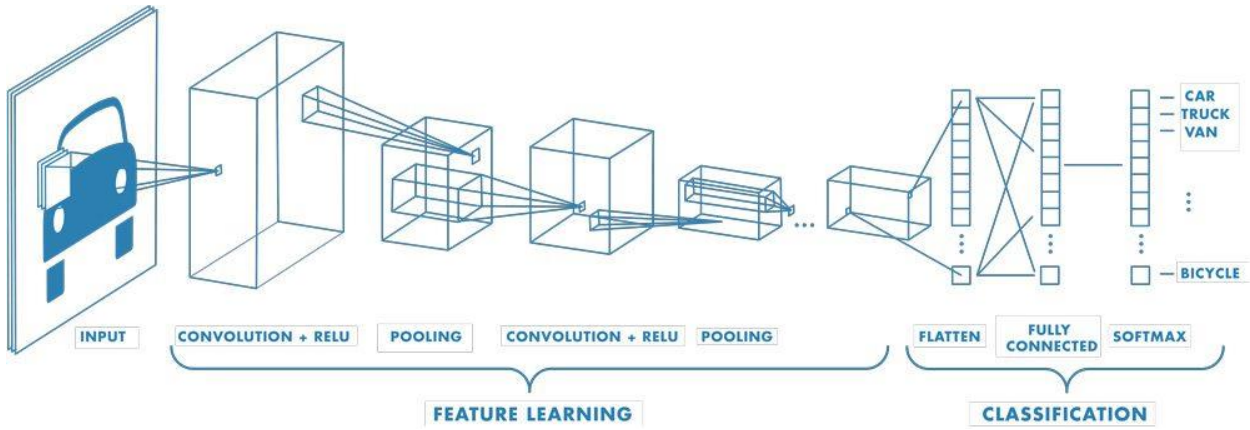
سبب عدم استخدام خوارزمية YOLO: لأنها أبطأ من SSD MobileNet وتحتاج أجهزة حسابية مكثفة ومعقدة.

المودل المستخدم مؤلف من شبكة أساسية MobileNet وشبكة اكتشاف SSD ومستخرج ميزات FPNLite

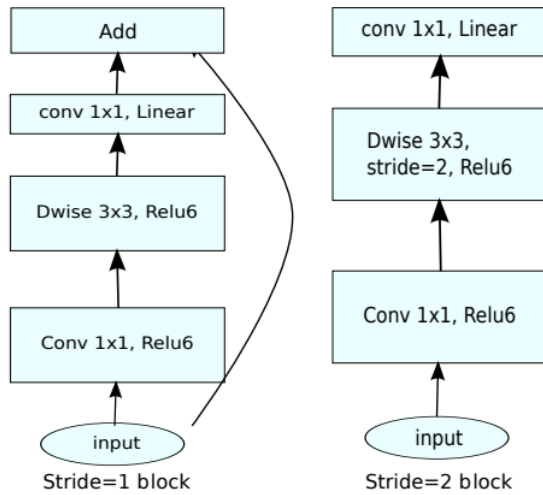
## أولاً : MobileNet:

شبكة عصبية توفر ميزات عالية المستوى للتصنيف classification أو الكشف detection للتصنيف: يمكن استخدام طبقة متصلة بالكامل fully connected و طبقة softmax بالنهاية





للكشف: يمكن إزالة الطبقة المتصلة بالكامل وطبقة softmax واستبدالها بشبكات الكشف مثل SSD ويسعى هذا النوع من الشبكات إلى الحصول على الأداء الجيد على الأجهزة المحمولة وهي مؤلفة من طبقة التفافية أولية كاملة initial fully convolutional layer مع 32 فلتر أو مرشح وتليها طبقة residual bottleneck



فيها نوعان من الكتل :

الأول هو residual block بخطوة 1

الثاني بخطوة 2 لتقليص الحجم

وكلا الكتلتين لهما 3 طبقات:

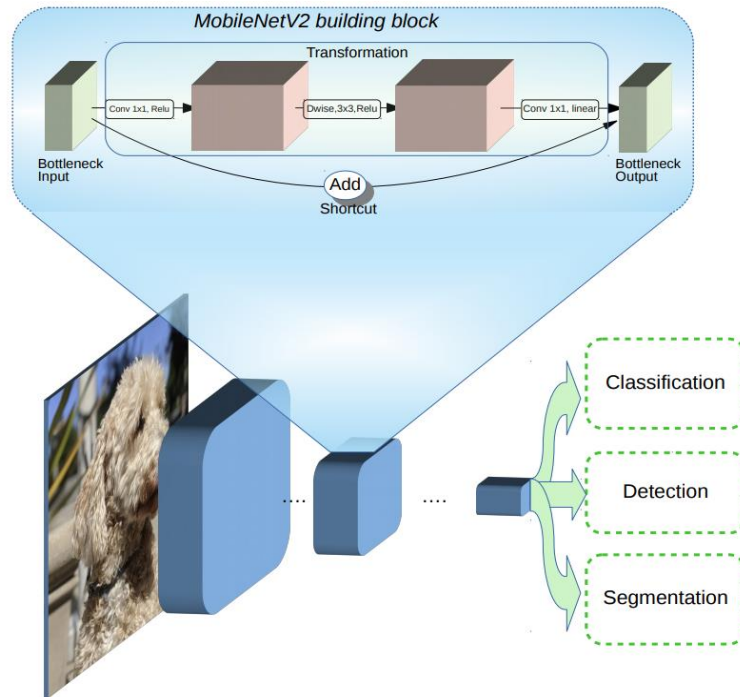
1. طبقة convolution 1x1 مع Relu6
2. Depthwise convolution
3. أيضاً convolution 1x1 لكن بدون خطية أي بدون Relu6 .

## (d) Mobilenet V2

ويكون الدخل والخرج للطبقات الثلاثة بالشكل:

Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

حيث t عامل تمدد ويكون مساوي 6 بجميع التجارب الرئيسية



ولقد استدمنا النسخة الثانية من هذه الشبكات MobileNet V2 لأنها تقدم ميزتين جديدتين:

1. الاختناقات الخطية بين الطبقات linear bottleneck:

تقوم بتشفير المدخلات والمخرجات الوسيطة للنموذج بينما تغلف الطبقة الداخلية قدرة النموذج على التحول من مفاهيم المستوى الأدنى مثل البكسل إلى المستوى الأعلى مثل فئات الصور

2. وصلات الاختصار بين الاختناقات:

تتيح تدريباً أسرع ودقة أفضل للنموذج.

وعند إقران هذه النسخة الثانية من الشبكة مع الكاشف SSD يكون الناتج أسرع بنسبة 35% تقريباً من النسخة الأولى للشبكة v1.



ويكون خرج هذه الشبكة هو متجه نموذجي يحتوي على بيانات الكائن

## ثانياً : Single Shot Detector SSD:

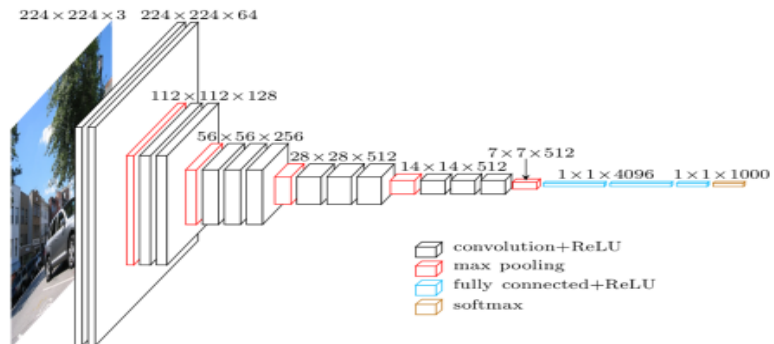
هو كاشف يمكنه التنبؤ بفئات متعددة فهو يكتشف الكائنات في الصور باستخدام شبكة عصبية واحدة عميقة عن طريق تحديد مساحة الإخراج للمربعات المحيطة في مجموعة من المربعات الافتراضية على نسب عرض إلى ارتفاع مختلفة ومقاييس لكل موقع ويقوم بإنشاء درجات لوجود كل كائن في كل مربع افتراضي ويضبط المربع ليلائم شكل الكائن بشكل أفضل وتجمع الشبكة أيضاً تنبؤات من خرائط معالم متعددة بدرجات دقة مختلفة للتعامل مع كائنات ذات أحجام مختلفة ولديها دقة أفضل من باقي الطريق وأسرع ويتميز عن غيره من الأجهزة بأنه يكتشف الكائنات في الزمن الحقيقي بسرعة ولكن قد يحدث مشكلة بالنسبة لكائنات ذات حجم أصغر.

يتكون اكتشاف الكائن باستخدام SSD من جزأين:

استخراج خرائط المعالم – تطبيق مرشحات أو فلاتر الالتفاف لاكتشاف الكائنات.

وهو يستخدم VGG16 لاستخراج المعالم ثم يكتشف الكائنات عبر طبقة Conv4\_3

حيث VGG16 هي عبارة عن بنية شبكة عصبية التلافيفية (CNN) وتعتبر واحدة من تصميمات نماذج الرؤية الممتازة حتى الآن. أكثر ما يميز VGG16 هو أنه بدلاً من وجود عدد كبير من المعلمات الفائقة ، فقد ركزوا على وجود طبقات التفاف من مرشح  $3 \times 3$  مع خطوة 1 واستخدموا دائماً نفس الحشو وطبقة الحد الأقصى للفلتر  $22 \times 22$  من الخطوة 2. ويتبع هذا الترتيب من الالتفاف وطبقات التجمع القصوى باستمرار. في النهاية يحتوي على 2 FC (طبقات متصلة بالكامل) متبوعاً بـ softmax للإخراج. يشير الرقم 16 في VGG16 إلى أنه يحتوي على 16 طبقة

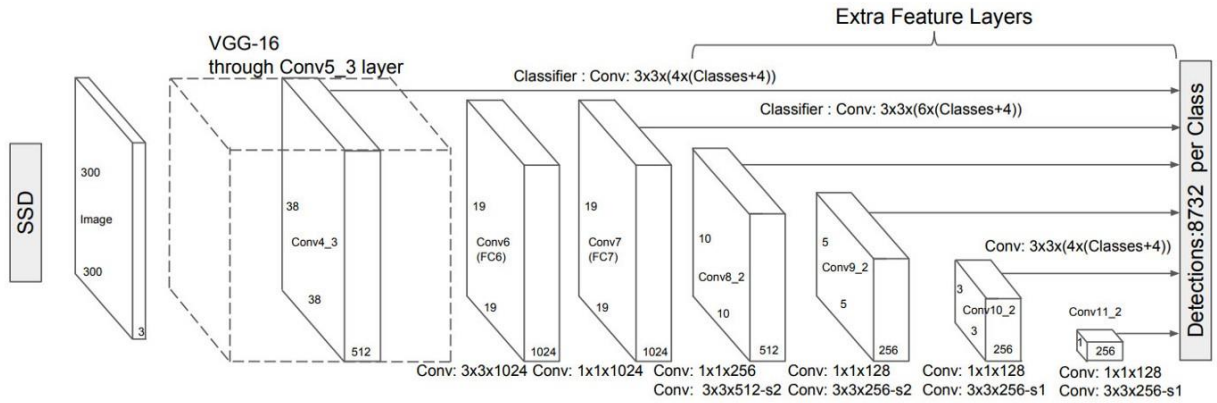


في العديد من التنبؤات من الممكن أن لا تحتوي على أي كائن لذلك يحتفظ SSD بفئة "0" للإشارة إلى عدم احتوائه على كائنات.

لا يستخدم SSD شبكة اقتراح منطقة مفوضة. بدلاً من ذلك ، يتم تحديد طريقة بسيطة للغاية. يقوم بحساب درجات الموقع والفئة باستخدام مرشحات الالتواء الصغيرة. بعد استخراج خرائط المعالم ، يطبق SSD مرشحات التفاف  $3 \times 3$  لكل خلية لعمل تنبؤات (ت حسب هذه المرشحات النتائج تمامًا مثل مرشحات CNN العادية).

وهو يستخدم طبقات متعددة (خرائط معالم متعددة المقاييس) لاكتشاف الكائنات بشكل مستقل. نظرًا لأن CNN تقلل البعد المكاني تدريجيًا ، فإن دقة خرائط الميزات تنخفض أيضًا لذلك يستخدم SSD طبقات دقة أقل لاكتشاف الكائنات ذات الحجم الأكبر.

ثم يضيف 6 طبقات التفاف إضافية CNN بعد VGG16 خمسة منهم لاكتشاف الكائنات.



توقع مربعات الحدود: تمامًا مثل التعلم العميق ، نبدأ بتنبؤات عشوائية ثم نستخدم النسب المتدرج (عرض إلى ارتفاع) لتحسين النموذج ولكن بدلاً من إجراء تخمينات عشوائية ، يمكننا بدء التخمينات اعتماداً على مربعات افتراضية والتي يتم اختيارها يدوياً. يحدد SSD قيمة مقياس لكل طبقة خريطة معالم. بدءاً من اليسار ، يكتشف Conv4\_3 الكائنات على أصغر مقياس ثم يزد خطياً إلى الطبقة الموجودة في أقصى اليمين بمقياس . بدمج قيمة المقياس مع نسب العرض إلى الارتفاع المستهدفة ، نحسب عرض المربعات الافتراضية وارتفاعها على النحو التالي:

$$w = scale \cdot \sqrt{\text{aspect ratio}}$$

$$h = \frac{scale}{\sqrt{\text{aspect ratio}}}$$

ثم يضيف SSD مربع افتراضي إضافي بقياس:

$$scale = \sqrt{scale \cdot scale \text{ at next level}}$$

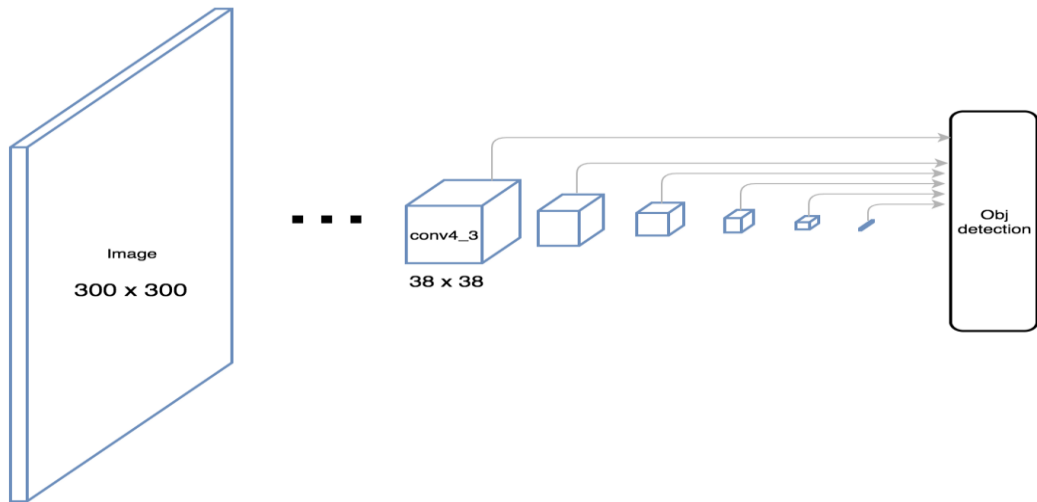
وحيث:

$$aspect \text{ ratio} = 1.$$

ثم يتم تصنيف توقعات SSD على أنها مطابقات إيجابية أو مطابقات سلبية بالشكل إذا كان مربع الحدود الافتراضي المقابل (وليس مربع الحدود المتوقعة) يحتوي على IoU أكبر من 0.5 مع الحقيقة الأساسية ، فإن المطابقة تكون إيجابية. خلاف ذلك ، فهو سلبي حيث  $IoU = \frac{\text{the intersection over the union}}$  التقاطع فوق الاتحاد: هو النسبة بين المنطقة المتقاطعة على المنطقة المرتبطة لمنطقتين.

فيتم تجاهل المطابقات السلبية ومتابعة العمل على المطابقات الإيجابية وبالتالي تشجع استراتيجية المطابقة هذه كل توقع على التنبؤ بأشكال أقرب إلى المربع الافتراضي المقابل.

تعد خرائط الميزات عالية الدقة مسؤولة عن اكتشاف الكائنات الصغيرة. الطبقة الأولى لاكتشاف الكائنات Conv4\_3 لها بُعد مكاني  $38 \times 38$  ، وهو تقليل كبير جدًا من صورة الإدخال. ومن ثم ، فإن أداء SSD عادة سيئ للأشياء الصغيرة مقارنة بطرق الكشف الأخرى. إذا كانت هناك مشكلة ، فيمكننا التخفيف من حدتها باستخدام صور ذات دقة أعلى

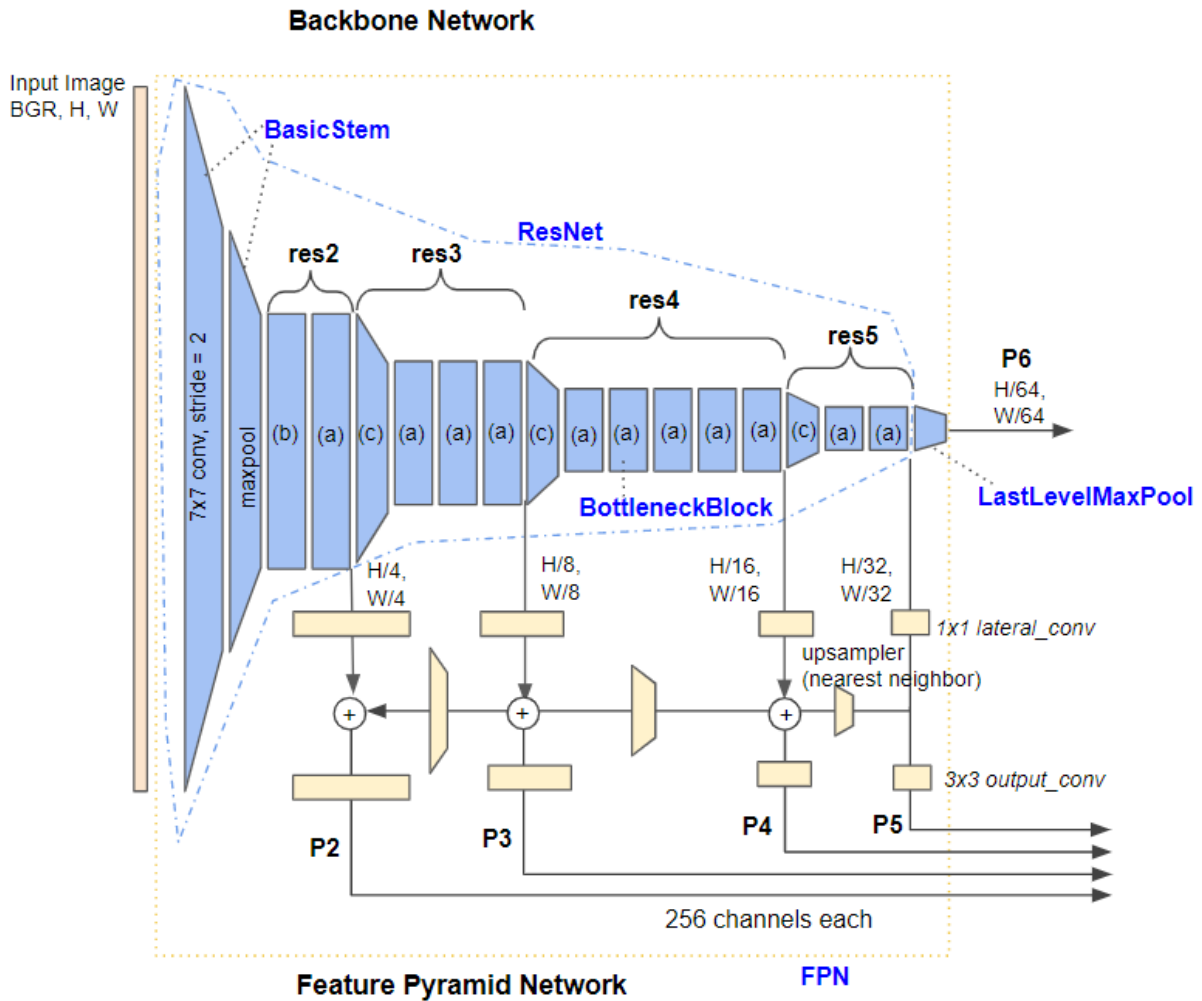


وفي حالة التوقعات المكررة لنفس الكائن : يستخدم SSD قمعًا غير أقصى لإزالة هذه التوقعات المكررة ويقوم بفرز التنبؤات حسب درجات الثقة بدءًا من أعلى تنبؤ للثقة ويقوم بتقييم ما إذا كانت أي مربعات حدود

تم توقعها مسبقاً تحتوي على IoU أعلى من 0.45 مع التنبؤ الحالي لنفس الفئة سيتم تجاهل التنبؤ الحالي. وبالتالي نحتفظ بأفضل 200 توقع لكل صورة على الأكثر.

### ثالثاً: Feature Pyramid Network FPN :

يعد اكتشاف الكائنات بمقاييس مختلفة أمراً صعباً بالنسبة للأجسام لذلك تعد FPN مستخرجاً للميزات لتحسين الدقة والسرعة وبذلك تنتج خرائط الميزات features maps لدقات مختلفة اعتماداً على الصورة المدخلة





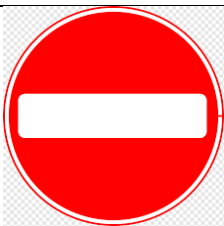
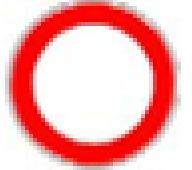
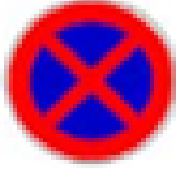
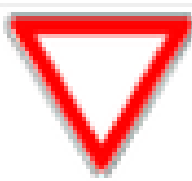
الدخل هو الصورة ( B , 3 , H , W )










H ارتفاع الصورة و W عرضها و B حجم الحزمة batch size وتكون الصورة المدخلة ملونة بالشكل BGR وليس RGB حيث أن إدخال صورة RGB قد يخفض دقة الكشف ويكون الخرج هو خرائط المعالم للكائن.







شبكة FPN تحتوي طبقة ResNet وطبقان convolution (إحدهما جانبية lateral والأخرى للخرج output) وآخذ العينات الأعلى up-samples وطبقة الحد الأقصى من المستوى الأخير last level maxpool .

### مجموعة البيانات المستخدمة Dataset:

استخدمنا داتا سيت مكونة من 6800 صورة واستخرجنا الفئات للإشارات الموجودة بكل صورة باستخدام الأداة labelmg ونتج لدينا 21 نوع من الإشارات المرورية وهي:

الفئة	مثال عن الفئة
CrossWalk	
children	
Don't enter	
Don't move	
Don't stop	
give road	

main road	
no overtaking	
no parking	
stop	
Speed limit 5	
Speed limit 10	
Speed limit 20	
Speed limit 30	
Speed limit 40	

Speed limit 50	
Speed limit 60	
Speed limit 70	
Speed limit 80	
Speed limit 90	
Speed limit 100	

على الرغم من العدد القليل للصور التي استخدمت للتدريب ولكن ذلك لم يشكل مشكلة لأن المودل المستخدم كان يقوم بعملية زيادة البيانات data augmentation والتي أدت إلى زيادة عدد الصور وجعلها أصعب.

بعد استخراج الفئات للإشارات من الصور:

تم تم إنشاء ملف CSV بناء على ملفات XML من مجموعة التدريب والاحتبار الناتجة عن الأداة labellmg حيث يحتوي هذا الملف على معلومات تتعلق بارتفاع الصورة وعرضها ونوع أو فئة الإشارة المكتشفة وعدد المربعات وإحداثيات كل مربع بالصورة ثم تم استخدام هذا الملف لتنسيق ملف TFRecord وهو ملف تخزين ثنائي خاص بشركة Tensorflow ويستخدم كأساس إدخال البيانات للتدريب والتحقق من صحة النتائج.

## تقييم المودل:

أعطى المودل دقة وسطية في اختبار جميع الحالات وصلت ل 82.14% .

وتم تقييم المودل باستخدام معدل الاسترجاع (الاستدعاء) recall الذي يعطي نسب الأشياء الفعلية التي التقطها المودل وهو يعطى بالعلاقة:

$$\text{Recall} = \text{TP}/(\text{TP}+\text{FN})$$

إضافة إلى معدل الدقة precision والذي يعطي نسبة الاكتشافات الصحيحة التي قام بها المودل وهو يعطى بالعلاقة:

$$\text{Precision} = \text{TP}/(\text{TP}+\text{FP})$$

حيث:

TP True Positive: اكتشاف صحيح للكائن بموقعه الصحيح

FP False Positive: اكتشاف صحيح للكائن لكن بغير موقعه

TN True Negative: عدم اكتشاف لكائن غير موجود في الصورة

FN False Negative: عدم اكتشاف لكائن موجود في الصورة

أعطى المودل معدل دقة وسطي  $\text{maP} = 0.81$

ومعدل استرجاع وسطي  $\text{mean recall} = 0.72$

## الصعوبات التي واجهتنا:

على الرغم من أن نفس النوع من إشارات المرور لها بعض الاتساق في اللون إلا أنه في البيئات الخارجية يتأثر لونها بشكل كبير بالإضاءة واتجاه الضوء وممكن أن يتشوه شكل الإشارة إذا كانت الكاميرا غير متعامدة معها أو كان السائق يتحرك بسرعة كبيرة بالإضافة إلى تأثير المباني والأشجار والمركبات الأخرى عليها فقد تحجبها في بعض المشاهد لذلك احتجنا إلى التعرف على الإشارات بمعلومات كاملة آخذين في الاعتبار تغير لونها وتأثير المطر والليل والضباب عليها وغيرها من المشاكل التي تعيق كشف الإشارات وتصنيفها.



## التنفيذ:



## الفصل السابع: الخاتمة

يهدف نظام تنبيه السائق الذكي الى مساعدة السائق وتزويده بالادوات المناسبة لتجنب الحوادث والمخالفات المرورية , كان هدف التطور في التكنولوجيا وما زال هو تحسين نوعية الحياة للبشرية , تم إنجاز النظام المقترح عن طريق تطبيق هاتف متجاوب ومتسق و تطبيق حاسب يمكن تضمينه داخل ال mini computers مثل ال raspberry Pi يساعد على إنجاز المهام المطلوبة والمتطلبات الوظيفية والغير وظيفية كما انه قابل للتطوير وإضافة ميزات و متطلبات إضافية بسبب الطريقة المتبعة في بنائه , كما أن لكل نظام او تطبيق عيوب مهما كان الفريق الهندسي والتقني الذي يعمل عليه فهناك أيضاً بعض العيوب والنواقص التي قد توجد داخل اي نظام , في النهاية لا يوجد نظام كامل خالي من العيوب لكن تقع الأهمية في بنائه بشكل صحيح . الاستمرار في تطوير أي منتج برمجي هو مفتاح نجاحه في أي سوق عمل . من الصعوبات التي واجهتنا خلال بناء النظام هي اختيار تقنية مناسبة من حيث السرعة والدقة وأيضاً القيام بعملية ال deployment والتي هي من اصعب مراحل بناء اي تطبيق ذكي بسبب نقص ال documentation عن الموضوع وصعوبة تنفيذ النظم على الاجهزة الضعيفة كالهاتف او ال Raspberry pi بسبب ضعف امكانياتها فهي لم تصمم من اجل ان تعمل على نماذج الذكاء الصناعي .

### مميزات النظام الذي تم بنائه:

- ✓ واجهة مستخدم مميزة وبسيطة
- ✓ كشف نعاس السائق عن طريق اكتشاف اغلاق العينين
- ✓ كشف نعاس السائق عن طريق اكتشاف التثاؤب
- ✓ اكتشاف اللافتات المرورية وتصنيفها
- ✓ تنبيه السائق في حالة اكتشاف النعاس او اللافتات المرورية

### مقترحات لتطوير النظام:

- 1- إضافة ميزة الاتصال بالنجدة او رقم محدد في حالة عدم ايقاف الانذار من قبل السائق .
- 2- إضافة نماذج رؤية حاسوبية اخرى تكشف عدم انتباه السائق في العديد من الحالات مثل التكلم عبر الهاتف او محاولة التقاط غرض من الخلف او انشغال السائق بالحديث مع الجالسين معه .
- 3- ربط النظام المقترح مع السيارة بشكل ألي مضمّن مما سوف يقلل الخطأ البشري بشكل كبير.
- 4- إضافة تصنيفات Classes اخرى للافتات المرورية التي يمكن للنظام اكتشافها .

[1] Yanzhao Zhu (07 March 2022).

[Traffic sign recognition based on deep learning | SpringerLink](#)

[2] Abdallah Benhamida & Miklos Kozlovsky(2020).

[Traffic Signs Recognition in a mobile-based application using TensorFlow and Transfer Learning technics | IEEE Conference Publication | IEEE Xplore](#)

[3] Irfan Kilic & Galip Aydin(2020).

[Traffic Sign Detection And Recognition Using TensorFlow' s Object Detection API With A New Benchmark Dataset | IEEE Conference Publication | IEEE Xplore](#)

[4] Grant Zhong (14 December 2019).

[Drowsiness Detection with Machine Learning | by Grant Zhong | Towards Data Science](#)

[5] Hitendra Garg (2020).

[Drowsiness Detection of a Driver using Conventional Computer Vision Application | IEEE Conference Publication | IEEE Xplore](#)

[6] [Image Classification vs. Object Detection vs. Image Segmentation | by Pulkit Sharma | Analytics Vidhya | Medium](#)

[7] [models/tf2\\_detection\\_zoo.md at master · tensorflow/models \(github.com\)](#)

[8] MediaPipe documentation

[MediaPipe](#)

[9] ML kit documentation

[Face Detection | ML Kit | Google Developers](#)