



جامعة دمشق

كلية الهندسة الميكانيكية والكهربائية

قسم الحواسيب والأتمتة

## تحويل لغة الإشارة إلى نص

مشروع تخرج أعدد لنيل إجازة في هندسة الحواسيب والأتمتة

إعداد الطلاب:

فرح شمس الدين

دينا الصلخدي

سعاد يعقوب آغا

محمد حسن الحلاق

بإشراف:

د.م. رؤوف حمدان

العام الدراسي

2019-2018

## فهرس الأشكال

- الشكل (1): نظام الإبصار عند الإنسان
- الشكل (2): مقطع عرضي في عين الإنسان يظهر العصي والمخاريط
- الشكل (3): العين المركبة
- الشكل (4): أقسام جزء الدماغ المسؤول عن الرؤية
- الشكل (5): مراحل معالجة الصورة الرقمية
- الشكل (6): الصورة المدخلة للبرنامج
- الشكل (7): الخرج التنفيذي لبرنامج قراءة صورة بتدرجات رمادية
- الشكل (8): الألوان الأساسية للصورة الملونة والألوان المشتقة من دمجها
- الشكل (9): تمثيل الصورة الرقمية في الحاسب
- الشكل (10): الخرج التنفيذي لتحويل لصورة ثنائية
- الشكل (11): نظام ألوان (HSV)
- الشكل (12): الفرق بين نظام الألوان (RGB) ونظام الألوان (CYMK)
- الشكل (13): معمارية الشبكة العصبونية الالتفافية
- الشكل (14): شبكة عصبونية التفافية بسيطة
- الشكل (15): عملية الالتفاف Convolution operation
- الشكل (16): تابع Relu
- الشكل (17): طبقة MaxPooling
- الشكل (18): تابع SoftMax
- الشكل (19): شرح طبقة Dropout
- الشكل (20): تمثيل بسيط لشبكة RNN
- الشكل (21): بنية LSTM
- الشكل (22): شعار نظام التشغيل Android
- الشكل (23): المكونات الأساسية لنظام الأندرويد
- الشكل (24): شعار برنامج Android Studio
- الشكل (25): متطلبات النظام الذي يحتاجه لتشغيل Android Studio
- الشكل (26): دورة حياة الفعالية
- الشكل (27): دورة حياة الخدمة
- الشكل (28): مقطع لبروتوكول التحكم بالنقل TCP
- الشكل (29): معمارية شبكة كشف الأحرف
- الشكل (30): منحني تغيرات الخطأ خلال التدريب

- الشكل (31): منحني Accuracy
- الشكل (32): معمارية شبكة التعرف على الإيماءات
- الشكل (33): منحني Accuracy
- الشكل (34): أيقونة تطبيق Android
- الشكل (35): الواجهة الرئيسية
- الشكل (36): واجهة البث
- الشكل (37): واجهة الإعدادات
- الشكل (38): واجهة تعريفية للبرنامج

## الفهرس

|   |  |
|---|--|
| فهرس الأشكال .....                                      |  |
| المصطلحات .....   |  |
| الفصل الأول : مقدمة .....                               |  |
| 1.1 مبررات المشروع .....                                |  |
| 2.1 فكرة عامة عن المشروع .....                          |  |
| 3.1 مقارنة بين المشروع الحالي و المشاريع المماثلة ..... |  |
| الفصل الثاني : معالجة الصورة و الإبصار الحاسوبي .....   |  |
| 1.2 مقدمة إلى الإبصار الحاسوبي .....                    |  |
| 2.2 الإبصار عند الإنسان .....                           |  |
| 3.2 العين المركبة .....                                 |  |
| 4.2 معالجة الصورة في الدماغ .....                       |  |
| 5.2 معالجة الصورة و الرؤية عند الحاسب .....             |  |
| 6.2 التقاط الصور و أنواع الكاميرات من حيث التقنية ..... |  |
| 7.2 التعامل مع الصور باستخدام مكتبة OpenCV .....        |  |
| 8.2 معلومات أساسية عن الصور .....                       |  |
| 9.2 أنظمة الألوان الأساسية .....                        |  |
| 10.2 مفهوم الدقة .....                                  |  |
| الفصل الثالث : الشبكات العصبونية الالتفافية CNN .....   |  |
| 1.3 العصبون .....                                       |  |

|     |   |
|-----|---|
| 2.3 | الشبكات العصبونية متعددة الطبقات              |
| 3.3 | التعلم و التدريب                              |
| 4.3 | بنية الشبكة العصبونية الالتفافية              |
| 5.3 | لمحة عن مكتبة TensorFlow                      |
| 6.3 | لمحة عن مكتبة Keras                           |
|     | الفصل الرابع : الشبكات العصبونية العودية RNN  |
| 1.4 | مقدمة في RNN                                  |
| 2.4 | استخدامات RNN                                 |
| 3.4 | مشكلة vanishing gradient                      |
| 4.4 | LSTM  |
| 5.4 | تدريب شبكة RNN                                |
|     | الفصل الخامس : مقدمة في الأندرويد             |
| 1.5 | تاريخ نظام Android                            |
| 2.5 | مميزات نظام Android                           |
| 3.5 | بنية نظام Android                             |
| 4.5 | بيئة Android Studio                           |
| 5.5 | مكونات تطبيق Android                          |
| 6.5 | ملف manifest file                             |
|     | الفصل السادس : بروتوكول الاتصال و قسم الإرسال |
| 1.6 | بروتوكول نقل النص التشعبي HTTP                |

|     |                                  |       |
|-----|----------------------------------|-------|
| 2.6 | بروتوكول التحكم بالنقل TCP       | ..... |
| 3.6 | بنية مقطع بروتوكول التحكم بالنقل | ..... |
|     | الفصل السابع : الإجراء العملي    | ..... |
| 1.7 | شبكة كشف الأحرف                  | ..... |
| 2.7 | شبكة التعرف على الإيماءات        | ..... |
| 3.7 | قسم الإرسال و تطبيق Android      | ..... |

## المصطلحات Glossary

|   |   |
|---|---|
| Android                                 | نظام تشغيل للأجهزة الذكية                         |
| Android Architecture                    | معمارية نظام التشغيل Android                      |
| IDE(integrated development environment) | بيئة تطوير نظام التشغيل Android                   |
| Open source                             | مفتوح المصدر                                      |
| Virtual machine                         | آلة عتادية افتراضية                               |
| Application components                  | مكونات التطبيق                                    |
| Manifest file                           | ملف البيان  |
| Pixel                                   | عنصر الصورة                                       |
| Detection/segmentation                  | الكشف/التجزئة                                     |
| OpenCV                                  | المكتبة البرمجية المفتوحة المصدر للرؤية الحاسوبية |
| HCI(Human computer interaction)         | أنظمة التفاعل بين الإنسان والحاسب                 |
| OOP(object oriented programming)        | البرمجة غرضية التوجه                              |
| Source Code                             | رماز المصدر                                       |

|                                   |                             |
|-----------------------------------|-----------------------------|
|                                   |                             |
| Object Source                     | رماز الغرض                  |
| Compiler                          | المترجم                     |
| RGB(Red-Green-Blue)               | نظام ألوان                  |
| BGR(Blue-Green-Red)               | نظام ألوان                  |
| HSV(Hue-Saturation-Value)         | نظام ألوان                  |
| CYMK(Cyan-Yellow-Magenta-Black)   | نظام ألوان                  |
| Dpi(dot per square inch)          | واحدة الدقة                 |
| CNN(Convolution Neural Network)   | الشبكة العصبونية الالتفافية |
| Artificial Neural Network         | الشبكة العصبونية الصناعية   |
| RNN(Recurrent Neural Network)     | الشبكة العصبونية العودية    |
| LSTM( Long short-term memory)     |                             |
| BPTT(BackpropagationThrough Time) | تدريب RNN في خطوة           |

|                                    |                        |
|------------------------------------|------------------------|
|                                    | زمنية معينة            |
| TCP(Transmission Control Protocol) | بروتوكول التحكم بالنقل |
| HLEN(Header Length)                | طول ترويسة             |

|                                   |                      |
|-----------------------------------|----------------------|
| APK(Application Programming Kite) | حزمة برمجة التطبيقات |
|-----------------------------------|----------------------|



# الفصل الأول

## مقدمة عن المشروع

## 1.1 مبررات المشروع:

لا يستطيع بعض البشر التواصل مع محيطهم بشكل تام لوجود عدد من المعوقات التي قد تكون خلقية أو نتيجة تعرضهم لحادث ما، وبالتالي يوجد كثير من الأشياء التي لا يستطيعون القيام بها أو وظائف لا يستطيعون الحصول عليها بسبب عدم مقدرتهم على التواصل، مثل: الصم، البكم، العمي.

ومن أجل مساعدة الصم والبكم في التواصل قام عدد من الناس في القرن الخامس عشر قبل الميلاد بابتكار لغة بسيطة خاصة بالبكم تمكن من إتقانها من التعبير عن نفسه واحتياجاته وتمكن غيره من فهمه ودخلت عليها الكثير من التطويرات إلى وقتنا هذا، ولكن لا يتعلم هذه اللغة إلا الذين يواجهون مثل هذه المشكلات أو الأشخاص الذين يريدون أن يتواصلوا معهم بطلاقة، هذا يعني أن عدد الأشخاص الذين يتقنون هذه اللغات يشكلون نسبة قليلة جدا من المجتمع قد لا تتجاوز 10%.

يوجد العديد من الصعوبات التي تواجه ذوي الاحتياجات الخاصة في حياتهم اليومية ولعل أبرزها وأصعبها بنفس الوقت، هو مشكلة التواصل مع الآخرين، فالكثير منا قد صادف إنسان أبكم ولم يستطع أن يفهم عليه أو أن يدرك ما يريد.

إن هذه المشكلة قد تسبب للشخص الشعور بالخوف أو الخجل فيحاول تجنب الحديث ما أمكن أو يقتصر حديثه فقط مع شخص يتقن لغته ويستطيع فهم ما يريد، وبالتالي إن هذه المشكلة تزيد من وحدتهم وانعزالهم عن المجتمع والمحيط الخارجي.

يُقال: " أن كل ذي عاهة جبار... "، يوجد هناك العديد من الشخصيات التاريخية العظيمة غيرت مجرى التاريخ وصنعت بصمات خالدة عبر العصور كانوا يعانون من عاهات أو إعاقات، لذلك لا يمكننا من الاستخفاف بذوي الاحتياجات الخاصة، لأن لو توفر لديهم الأدوات المناسبة والتقنيات التي تساعد على التغلب على إعاقاتهم وتسهل تواصلهم مع المجتمع لربما صنعوا الكثير من الإنجازات أو اكتشفوا أشياء جديدة، فمن يعلم!

وهنا أتت فكرة المشروع، فعندما نقدم تطبيق للإنسان الأبكم نكون قد حللنا مشكلة رئيسية في حياته اليومية، والتي هي مشكلة التواصل مع المحيط، وبدورها سوف تحل العديد من المشاكل الأخرى.

## 2.1 فكرة عامة عن المشروع:

يقوم مشروعنا بحل المشكلات التي سبق ذكرها عن طريق تقديم تطبيق للهاتف النقال لتحويل لغة الإشارة المعنية بالأمور الصحية و الطبية (للاستخدام في المستوصفات و المشافي و المخابر ...) إلى نص مكتوب، حيث يقوم التطبيق بأخذ صور من كاميرا الهاتف النقال و إرسال هذه الصور الى جهاز حاسب يعمل كعميل client لمعالجة هذه الصور و تحليل الحركة في الزمن الحقيقي عن طريق شبكة عصبونية مدربة على بعض حركات لغة الصم و إظهار نتيجة الاختبار كنص مكتوب على الحاسب .

## 3.1 مقارنة بين المشروع الحالي والمشاريع المماثلة:

### نقاط ضعف المشاريع السابقة:

- اعتمدت بعض المشاريع المشابهة لمشروعنا على استخدام قفاز للتعرف على حركة اليد وتحليلها لاتخاذ قرار الكلمة المنطوقة المقابلة لهذه الحركة، وبهذه الحالة سوف يضطر الشخص الأبكم لشراء هذا القفاز ليتمكن من التواصل مع المحيط ومن الممكن ان يكون باهظ الثمن و كانت احدى المشاكل التقنية التي تواجههم :
- استخدام المبدل التماثلي الرقمي ADC لقراءة قيم الحساسات الموجودة على القفاز وكانت الصعوبة في التغير السريع في قراءة قيم المبدل ADC
- استخدام Bluetooth module للربط بين Microcontroller الموجود في القفاز وجهاز Android وكان هناك صعوبة في المزامنة بين طرفي الإرسال ( Bluetooth module من جهة، والموبايل من جهة أخرى)

- و كما اعتمدت مشاريع أخرى على معالجة الصورة و استخدام خوارزميات الالبصار الحاسوبي للتعرف على حركة اليد و تحليلها لاتخاذ قرار الكلمة المقابلة لهذه الحركة , و في هذه الحالة سيكون هناك العديد من المحددات و الشروط التي يتوجب تحققها جميعها من اجل الحصول على تحليل ممتاز للحركة :
  - دقة الكاميرا جيدة.
  - ظروف الإضاءة مناسبة.
  - يجب أن تكون اليد باتجاه معين أو بزاوية معينة.
  - عزل اليد عن المحيط من أجل تمييزها ومعرفة الحركة.
- وبالتالي نجد الكثير من الصعوبات تواجهنا في استحصال الحركة بشكل مثالي عند استخدام معالجة الصورة و خوارزميات الالبصار الحاسوبي التقليدية , هذا الأمر يجعل المشروع غير عملي ومن المستحيل أن يتم تطبيقه على أرض الواقع.
- مميزات مشروعنا :
  - الاعتماد على تدريب شبكة عصبونية على عدد من كلمات لغة الإشارة في بيئات وظروف إضاءة مختلفة مما يزيد من دقة النتائج
  - الاستفادة من الخدمات التي تقدمها لنا أنظمة تشغيل الأجهزة الذكية والتي تعتبر متوفرة لجميع الناس في وقتنا الحالي، استفدنا من الخدمات التالية:
    - Socket service
    - قوة المعالج وسرعته
  - يتم تبادل المعلومات بين الحاسب وجهاز الموبايل بشكل لاسلكي باستخدام شبكة LAN

- سهل الاستخدام
- حجم التطبيق صغير مقارنة بالتطبيقات الأخرى





الفصل الثاني  
معالجة الصورة و الإبصار  
الحاسوبي  
Image processing and  
computer vision

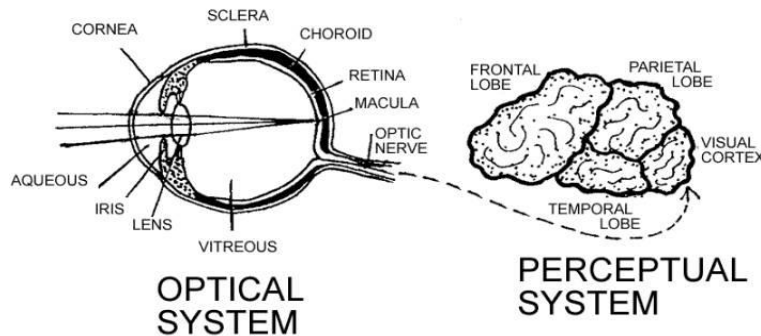


### 1.3 مقدمة إلى الإبصار الحاسوبي :

من الممكن وصف الرؤية الحاسوبية باعتبارها مرادفاً - وليس بالضرورة عكساً- للرؤية الفيزيولوجية. فكما أن الرؤية الفيزيولوجية للإنسان والحيوانات المختلفة تتم دراستها للتعرف على خصائصها، فإن علم الرؤية الحاسوبية يدرس ويصف أنظمة الرؤية الصناعية التي يتم تنفيذها في البرامج أو الأجهزة. وقد أظهر التعاون بين مجالي دراسة الرؤية الفيزيولوجية والحاسوبية تطوراً في تعميق الفهم لكلا المجالين.

### 2.3 الإبصار عند الإنسان :

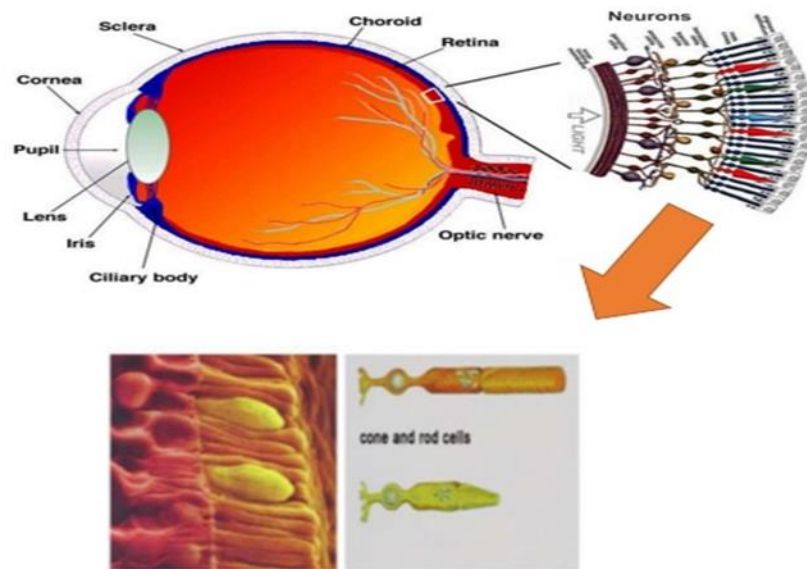
تعتبر العين المرحلة الأولى من نظام الإبصار عند الإنسان كما في الشكل (2-1) والعين مكونة من أقسام عديدة - والتي لها دورٌ هام في عملية ضبط شدة (Iris) كما يوضح الشكل (2-2) أهمها القرنية الشفافة، والقزحية الداخلية للعين والتي تضيق كلما ازدادت شدة الإضاءة وتتوسع كلما نقصت شدة الإضاءة، فهي تقوم بعملية ضبط أوتوماتيكي لشدة الإضاءة (auto - brightness - adjuster) والتي تضيق تارةً وتتوسع تارةً أخرى لجعل الصورة أكثر وضوحاً ويدخل الضوء بعد ذلك إلى الحدقة في الكاميرات الرقمية الحديثة، ينتقل الضوء بعد ذلك إلى الشبكية والتي عليها (auto focus) وهي تقابل توضع عصبونات خاصة حساسة للضوء تسمى المخاريط (cones) الحساسة للون والعصي (rods) الحساسة للضوء حيث تقوم هذه العصي بتحويل الضوء الوارد إلى إشارة كيميائية بعمليات معقدة تنتقل عبر العصب البصري إلى الدماغ حاملة معلومات عن النموذج الذي رسمته على شبكيتها،



### - VISUAL SYSTEM -

الشكل(1): نظام الإبصار عند الإنسان

في الحقيقة إنّ العصي والمخاريط في العين تقابل العناصر التي تكوّن الصورة في الكاميرات والتي تسمى pixel ولكن في الكاميرا تلك العناصر متماثلة وتسمى (photo resistor) في حين أنها غير متماثلة في العين فهناك اختلاف بين العصي والمخاريط ، تعمل العصي في الضوء الخافت حيث تعمل على توفير رؤية الأبيض والأسود و هي أكثر عدداً و تتركز في أطراف الشبكية. أما المخاريط فوظيفتها دعم الرؤية بالنهار وإدراك باقي الألوان و تتركز في مركز الشبكية.



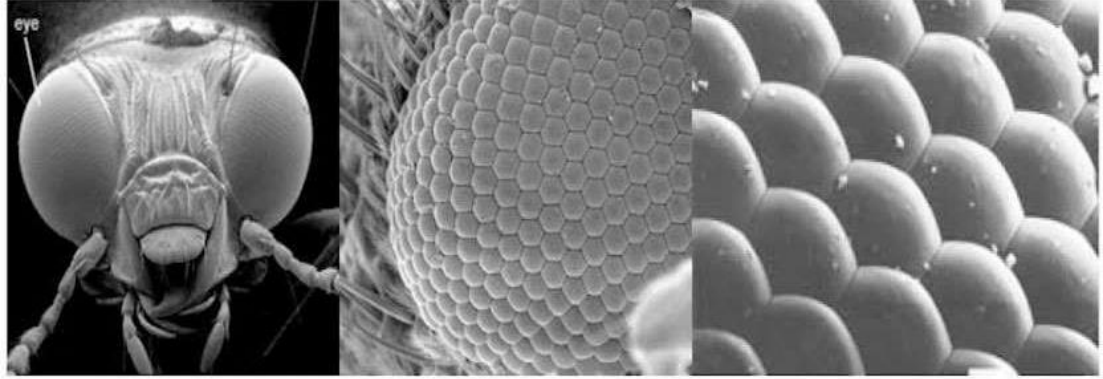
الشكل(2): مقطع عرضي في عين الإنسان يظهر العصي والمخاريط

### 3.3 العين المركبة :

هذا النوع من العيون يعمل بآلية مشابهة لعين الإنسان إلا أن كل عين بسيطة ( عينة ) من العين المركبة تمثل عنصراً من عناصر الصورة أي (pixel) على خلاف العين البشرية كما في الشكل التالي، ففي الواقع إن وجود المئات من العيون البسيطة مكوّنة عيناً مركبة واحدة في الحشرات لا يعني أن الحشرة ترى مئات الصور بل صورة وحيدة ولكن مكوّنة من المئات من العناصر(pixel).

دقة رؤية العين المركبة محدودة بسبب العدد الكبير للعينات وهي أقل بكثير من دقة

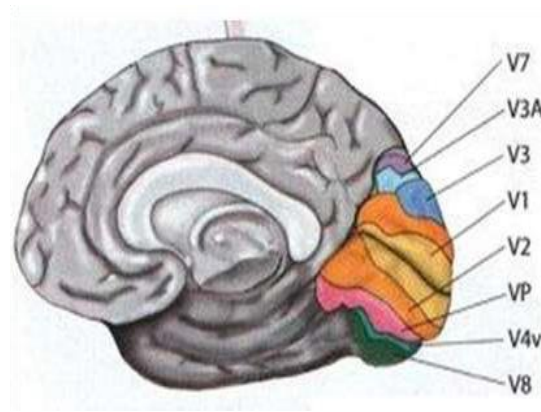
رؤية العين العادية ذات العدسة . ولكن سرعة الرؤية في العين المركبة تكون أسرع بكثير من سرعة رؤية العين البشرية. وهناك روبوتات مزودة بعيون مركبة مؤلفة من العديد من المقاومات الضوئية photo resistors والتي تربط مع بعضها لتشكل صورة رمادية بسيطة.



الشكل (3): العين المركبة

### 4.3 معالجة الصور في الدماغ:

الدماغ هو المكان حيث يتم تحويل المعلومات المتعلقة بالصورة إلى شيء مفهوم ثم إلى صورة، طبعاً إن الجزء المسؤول عن الرؤية في الدماغ مقسم إلى عدد من الأقسام كل منها يقوم بعملية معالجة مستقلة عن القسم الآخر ( Separate Subsystems) فمثلاً هناك جزء مسؤول عن تحديد الحركة وآخر عن معالجة الألوان وجزء مسؤول عن تحديد الشكل وأبعاده والشكل التالي يظهر تلك الأقسام.



الشكل(4): أقسام جزء الدماغ المسؤول عن الرؤية.

### 5.3 معالجة الصور والرؤية عند الحاسب :

تختلف أنظمة الرؤية الحاسوبية بشكل كبير وتتوزع بين أنظمة كبيرة ومعقدة تؤدي مهام عامة وشاملة، وبين أنظمة صغيرة تؤدي مهام مخصصة وبسيطة. ولكن معظم أنظمة الرؤية الحاسوبية تشمل العناصر التالية بشكل أساسي:

#### • الحصول على الصورة:

يتم الحصول على الصورة باستخدام واحد أو أكثر من مستشعرات الصور، وهذه تتضمن العديد من كاميرات مستشعرات الضوء، مستشعرات المسافات، أجهزة التصوير الشعاعي، الرادار، كاميرات الموجات فوق صوتية..الخ. وتبعاً لنوع المستشعر فإن الصورة الناتجة تكون ثنائية البعد أو ثلاثية البعد أو سلسلة صور متعاقبة. تكون قيمة كل بكسل في الصورة تابعة لقيمة شدة الإشعاع الضوئي في واحد أو أكثر من الحزم الضوئية (الصور الرمادية، أو الصور الملونة) ولكن أيضاً من الممكن أن تشير إلى العديد من القياسات الفيزيائية كالبعد، الامتصاص، أو انعكاس الموجات الكهرومغناطيسية.

#### • العمليات المسبقة:

قبل تطبيق خوارزمية الرؤية الحاسوبية على بيانات الصورة من أجل الحصول على معلومات مفيدة، فإنه من الضروري إجراء عمليات مسبقة على البيانات من أجل تأكيد أن البيانات تحقق افتراضات محددة تابعة للخوارزمية، بعض الأمثلة على هذه العمليات هي: إعادة تحديد دقة الصورة من أجل تأكيد صحة نظام إحداثيات الصورة. التقليل من التشويش من أجل التأكد أن المستشعر لا يقوم بتقديم أي معلومات خاطئة.

زيادة التباين من أجل التأكد من أن المعلومات المرغوبة سيكون من الممكن الحصول عليها.

#### 1. استحصال معالم الصور Feature extraction :

يتم الحصول على معالم الصورة بمستويات دقة مختلفة من بيانات الصورة ذاتها. تصنف هذه المعالم إلى:

- معالم عامة global features مثل اللون والشكل.

• معالم محلية local features كالزوايا Harris corner ، والبقع و SIFT features

ومن الممكن الحصول على معالم معقدة أكثر متعلقة بالألوان والأشكال في الصورة.

2. الكشف-التقسيم Detection/Segmentation: يتم تحديد مناطق هامة من الصورة من أجل العمليات اللاحقة.

مثلاً: اختيار مجموعة من نقاط العلام المميزة، تقسيم صورة أو أكثر تحتوي على المنطقة التي تحتوي الجسم المهتم به.

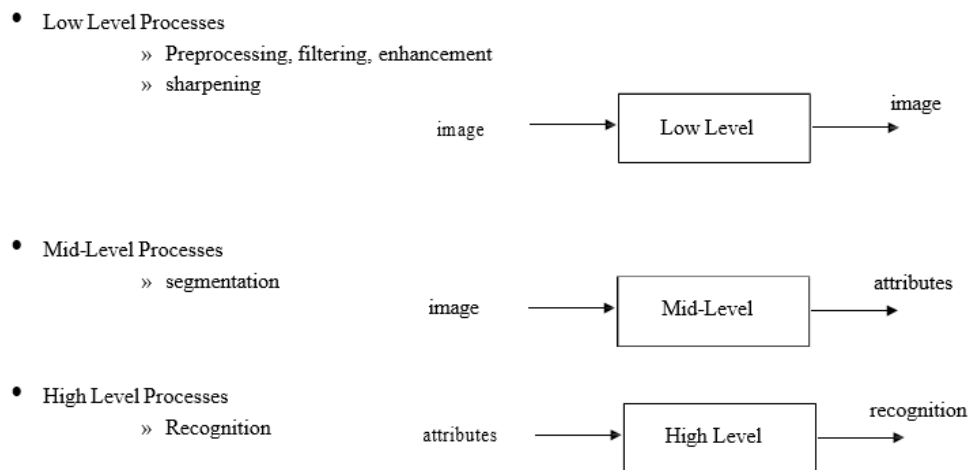
3. العمليات عالية المستوى: عند هذه المرحلة تكون البيانات المدخلة هي مجموعة صغيرة من البيانات، على سبيل المثال مجموعة من النقاط أو منطقة من الصورة التي يشك أنها تحتوي الجسم قيد الدراسة. والعمليات المتبقية تقوم بما يلي:

• التأكد من أن البيانات التي تم الحصول عليها توافق افتراضات التطبيق المقترح.

• تقدير قيم المعاملات المعينة للتطبيق، كاتجاه الجسم أو حجم الجسم.

• تصنيف الأجسام التي تم التعرف عليها في عدة فئات.

إذا يمكننا تقسيم مراحل معالجة الصورة الرقمية لثلاث مستويات كما يوضح الشكل التالي:



الشكل(5): مراحل معالجة الصورة الرقمية

### 6.3 التقاط الصور وأنواع الكاميرات من حيث التقنية :

الخطوة الأولى لمعالجة الصورة في أي نظام إبصار هي التقاط الصورة كما تحدثت في الفقرة السابقة ويتم ذلك من خلال الكاميرا (مستشعرات الصور image sensors)

التي تعتبر بمثابة العين عند الإنسان وتقسم الكاميرات بدورها إلى كاميرات من نوع (CCD: Charge-Coupled-Device) و كاميرات رقمية من نوع (Complementary metal-oxide-semiconductor :CMOS).

إن الكاميرات من نوع CCD تلتقط الصورة من خلال تحويل الطاقة الضوئية إلى شحنات كهربائية ثم تحويل تلك الشحنات الكهربائية حسب شدتها إلى إشارات رقمية باعتماد (ADC: Analog To digital Converter).

بينما الكاميرات من نوع CMOS فيها العديد من الترانزستورات لكل بكسل في الصورة وبالتالي الإشارة الناتجة هي إشارة رقمية ولا حاجة لمبدل تشابهي رقمي وتعتمد دقة الصورة على تقنية الترانزستورات.

إن الحساسات في الكاميرات من نوع (CCD) تعطي صوراً أكثر وضوحاً وأكثر دقةً من الكاميرات الرقمية (CMOS) والتي تكون حساساتها عرضة للضجيج بشكل أكبر.

بسبب وجود العديد من الترانزستورات بالقرب من كل بكسل في كاميرات CMOS وهذا يؤدي إلى أن العديد من الفوتونات تضرب الترانزستورات بدلاً من الثنائي الضوئي مما يؤدي لتقليل دقة الصورة.

الكاميرات من نوع CMOS تستهلك طاقة أقل بكثير من الكاميرات من نوع CCD .

إن الطاقة التي تستهلكها كاميرا من نوع CCD تزيد حوالي 100 مرة عن تلك المستهلكة في كاميرات من نوع CMOS .

## 7.6 التعامل مع الصور باستخدام لغة python ومكتبة OpenCV :

يعد التعامل مع الصور باستخدام لغة بايثون أمراً يسيراً نتيجة التوابع المتاحة من المكتبات المطورة في هذا المشروع تم استخدام المكتبة البرمجية المفتوحة المصدر للرؤية الحاسوبية (OpenCV) .

### لمحة تاريخية عن مكتبة OpenCV

المكتبة البرمجية المفتوحة للرؤية الحاسوبية (OpenCV) مكتبة برمجية تهدف بشكل أساسي لتطوير الرؤية الحاسوبية ، طورت من قبل شركة انتل (Intel). وهي مجانية تحت رخصة المصدر المفتوح (open source BSD license). ومدعومة من كافة أنظمة التشغيل ( Windows, Linux, Mac OS, iOS and Android )، مكتوبة بلغة C/C++ مما يعطيها أداء وسرعة كبيرين فهي تركز بشكل أساسي على المعالجة في الزمن الحقيقي (real-time).

أطلقت بشكل رسمي عام 1999م وهي قيد التطوير حتى الآن آخر إصدار لها OpenCV3.X

### تطبيقات مكتبة OpenCV :

من أهم التطبيقات التي تستخدم فيها المكتبة:

- إنشاء أدوات الرسم الثنائي والثلاثي الأبعاد.
  - أنظمة التعرف على الحركة
  - أنظمة التعرف على الوجوه.
  - أنظمة الروبوتات النقالة.
  - أنظمة كشف الحركة.
  - أنظمة التفاعل بين الإنسان والحاسب Human-computer interaction (HCI).
  - التعرف على الإيماءات .
- كما وأن مكتبة OpenCV تستخدم بكثرة مع خوارزميات تعليم الآلة والشبكات العصبونية.

### 8.3 معلومات أساسية عن الصور:

الصورة الرقمية مجموعة من العناصر الصغيرة تسمى البكسلات (pixels) يحمل كل بكسل قيمة تعبر عن هذا العنصر في الصورة، وتُخزن الصورة على شكل مصفوفة ثنائية البعد ( $n*m$ ) حيث  $n$  و  $m$  هو عدد البكسلات المكونة للصورة شاقولياً وأفقياً بالترتيب.

تُخزن الصورة الرقمية في الحاسب بعدة لواحق من أهمها: BMP، HDf JPEG، XWB، TIFF، PCX.

### 9.3 أنظمة الألوان الأساسية :

كي نتمكن من التعامل مع الصور في برنامجنا لابد من قراءتها أولاً ثم تحويلها لصيغة معينة حسب متطلبات العمل. ومن الصيغ المتاحة لنا التعامل معها:

#### • الصورة الرمادية المتدرجة (Gray scale Image):

يكون لكل بكسل من بكسلات الصورة قيمة تعبر عن مدى درجة اللون الرمادي.

ويمكن تمثيل الصورة ذات التدرج الرمادي بنوعين من الأعداد. الأول هو أعداد عشرية بين 0 الذي يمثل اللون الأسود و 1 لتمثيل اللون الأبيض مثل (double) وباقي القيم هي أعداد عشرية بين 0 و 1.

والنوع الثاني من الأعداد هي الأعداد الصحيحة بدون إشارة بطول 8-بت (uint-8) حيث يجري تمثيل تدرج الرماديات بين 1 لتمثيل اللون الأسود و 255 لتمثيل اللون الأبيض حيث الألوان القريبة من اللون الرمادي الفاتح تكون قريبة للعدد 255 والأرقام القريبة للرمادي الغامق تقترب من 1.

والمثال التالي يوضح قراءة صورة كتدرج رمادي وطباعة قيم البكسلات، حيث يمثل كل سطر بالمصفوفة بقائمة [ list ] ضمن المصفوفة المكونة للصورة.

رماز البرنامج:



```

import numpy as np
import cv2
#read image
image=cv2.imread('background.jpg',0)
print(img)
#show true color image
cv2.imshow('gray',image)
i=cv2.imdecode(image,-1)
print(i)
Esc=cv2.waitKey(0)
#if press 'q' quite program
if Esc == 27:
    cv2.destroyAllWindows()
elif Esc==ord('s'):
    #store image if press 's' key
    cv2.imwrite('gray.jpg',img)
    cv2.destroyAllWindows()

```



الشكل(6): الصورة المدخلة للبرنامج

الخرج التنفيذي:



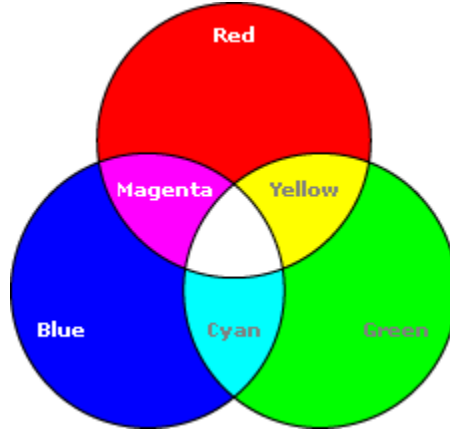
```
[[246 250 253 ..., 21 22 25]
[249 251 252 ..., 15 16 19]
[253 252 251 ..., 17 17 16]
...,
[ 33 34 36 ..., 249 251 252]
[ 33 34 36 ..., 247 249 251]
[ 33 34 36 ..., 250 250 251]]
```

الشكل(7): الخرج التنفيذي لبرنامج قراءة صورة بتدرجات رمادية

## • الصورة الملونة RGB Image :

تنتج الصور الملونة من مزج نطاقات (أقنية) ذوات أطوال موجية مختلفة من خلال إسقاطها عبر منابع ضوئية للألوان الأساسية الثلاثة. وأما بيانات هذه الصور فتكون مخزونة على شكل مصفوفة ثلاثية الأبعاد  $(M \times N \times 3)$ ، إذ يمثل البعد الأول اللون الأحمر (Red) والبعد الثاني يمثل اللون الأخضر (Green) أما البعد الثالث فيمثل اللون الأزرق (blue)، وكل بعد يتكون من  $(M \times N)$  من الأسطر والأعمدة وقيمة الشدة اللونية لكل لون في موقع النقاط الصورية المقابلة هي نفسها عند كل بعد، ويعد هذا النوع من الصور جيداً لتغطية مدى كامل من الألوان التي تتحسسها العين البشرية والذي يؤدي إلى إمكانية التمييز الأفضل للتشكيلات والأجسام المحتوات في الصورة وفي عمليات التحديد والتمييز الأفضل للعينات الطيفية التي تستخدم في تصنيف الصور. إلا أن هناك بعض المساوئ باستخدام هذا النوع من الصور إذ يكون مكلفاً حسابياً وفي بعض الأحيان ليس من الضروري استخدام (24-bits) لكل نقطة في الصورة ويحتاج إلى ذاكرة كبيرة ووقت للمعالجة لتخزين الصور الرقمية الملونة. أي إن كل بكسل يتم تمثيلها بثلاثة ألوان وكل لون يتم تمثيله بثمانية أرقام وهذا يعني أن كل نقطة صورية من نوع (RGB) سيتم تمثيلها من خلال أربعة وعشرين رقماً (24-bit/pixel).

يستخدم نظام الألوان RGB في الإظهار كما في شاشات الحاسوب والأجهزة الذكية. ولكن تعتمد المكتبة البرمجية المفتوحة المصدر للرؤية الحاسوبية (OpenCV) صيغة الألوان BGR وهو ذاته مع عكس القناتين للونين الأزرق والأحمر. نحتاج للتحويل من صيغة BGR إلى RGB القيام باستخلاص الأقنية الثلاث ومن ثم إعادة دمجهم من جديد في حال تعاملنا مع مكتبة أخرى تتعامل مع الصيغة RGB. ونحن لسنا بحاجة إلى ذلك في حال تعاملنا فقط مع توابع هذه المكتبة.



الشكل(8): الألوان الأساسية للصورة الملونة والألوان المشتقة من دمجها.

رماز البرنامج :

```

import numpy as np
import cv2
#read image
img=cv2.imread('background.jpg',-1)
print(img)
#show true color image
cv2.imshow('image',img)
i=cv2.imdecode(img,-1)
print(i)
Esc=cv2.waitKey(0)
#if press 'q' quite program
if Esc == 27:
    cv2.destroyAllWindows()
elif Esc==ord('s'):
    #store image if press 's' key
    cv2.imwrite('img_save.jpeg',img)
cv2.destroyAllWindows()

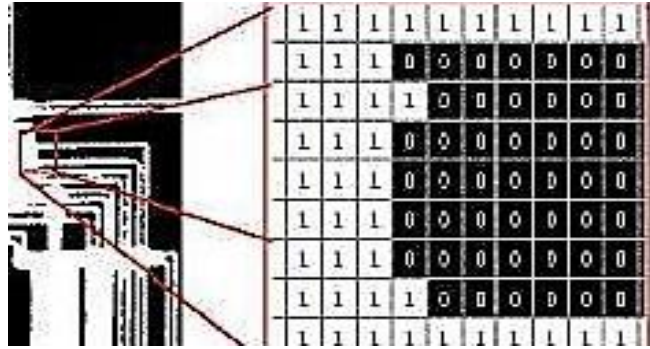
```

## • الصورة الثنائية : Binary Image

تعد الصور الثنائية أبسط أنواع الصور، إذ يتم استخدام التمثيل الرقمي الثنائي للبيانات (0,1) ، ويمكن أن يعبر عنها بالدالة  $F(i,j)$  وهي ثنائية الأبعاد، إذ يمثل (i) الإحداثية الأفقية أما (j) فيمثل الإحداثية العمودية ونقطة الأصل (0,0) تقع عند الزاوية العليا اليسرى من الصورة. ويتم تمثيل الأجسام أو الأشياء (Foreground) التي تحتويها الصورة بالقيمة (0)، أو إن قيم نقاط الصور الثنائية لها احتمالان اثنان يظهران بصورة طبيعية (Normally) كأسود وأبيض، وبصورة عددية (0) إذ القيمة (0) تمثل اللون الأسود والقيمة (1) تمثل اللون الأبيض.

و تستخدم الصورة الثنائية بكثرة في معظم خوارزميات معالجة الصورة الرقمية.

والشكل التالي يوضح توزيع البكسلات في صورة ثنائية.



الشكل (9): تمثيل الصورة الرقمية في الحاسب

رماز البرنامج:

```
import cv2
import numpy
#read image in grayscale level
Pic=cv2.imread('background.jpg',0)
#Pic =cv2.cvtColor(Pic,cv2.COLOR_RGB2GRAY)
ret,imageBinary=cv2.threshold(Pic,thresh=80,maxval=255,type=cv2.THRESH_BINARY)
cv2.imshow("GRAY_IMAGE",Pic)
cv2.imshow('BINARY_IMAGE',imageBinary)
cv2.imwrite('Pic.jpg',Pic)
cv2.imwrite('imageBinary.jpg',imageBinary)
cv2.waitKey(0)
```

الخرج التنفيذي:



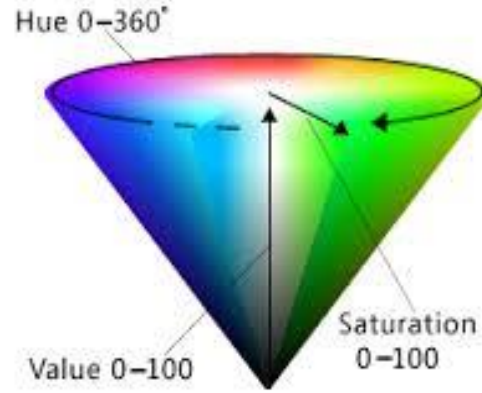
الشكل (10): الخرج التنفيذي للتحويل لصورة ثنائية

يوجد أنظمة ألوان أخرى مثل نظام الألوان HSV وهو مطابق تماماً لاستخدامات نظام الألوان RGB

ولكن يختلف بطريقة تمثيله للألوان . وترميز المحارف على التالي : Hue- Saturation-Value

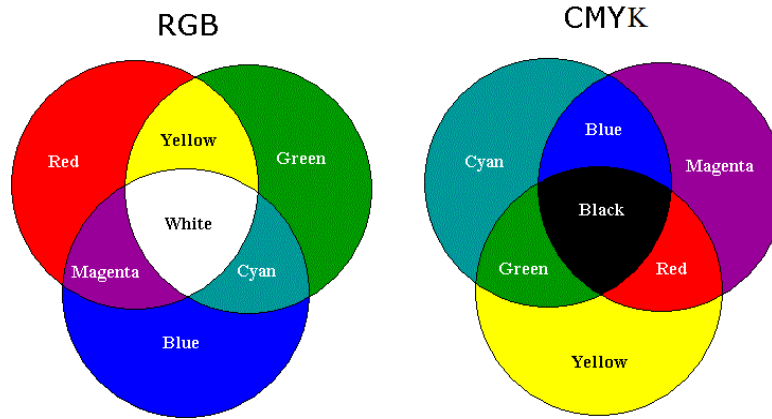
حيث تمثل الألوان في مخروط قاعدته تعبر عن درجة اللون (Hue) وهي بين 0° و 360° و نصف قطر قاعدة المخروط تعبر عن نسبة إشباع (Saturation) اللون بين 0% إلى 100%، بينما قيمة اللون التي تعبر عن نسبة الإضاءة (Brightness) للون بين 0% إلى 100%، فهي تمثل بارتفاع المخروط ويظهر الشكل (10-2) تمثيل الألوان بنظام HSV.

ملحوظة: إن استخلاص القناة H لوحدها وعرضها يعطينا صورة ذات تدرجات رمادية.



الشكل (11): نظام ألوان HSV

ويوجد أيضاً نظام ألوان الطباعة CMYK و هو مستخدم لأنه لعدم وجود لون أحمر صافي في الطبيعة وترميز الأحرف تعبر على التالي: Cyan-Yellow-Magenta-Black  
لم يستخدم الحرف B ليعر عن اللون الأسود وأستخدم بدلاً عنه الحرف K للفرقة لا أكثر.



الشكل (12): الفرق بين نظام الألوان RGB ونظام الألوان CMYK

### 10.3 مفهوم الدقة :

عندما تخزن الصورة الرقمية في الحاسب فهي تخزن على شكل ملف يحوي معلومات تفصيلية عن كل بكسل في الصورة من حيث اللون والموضع، ولكل صورة عدد ثابت من البكسلات في واحدة الحجم وهذا ما يسمى بدقة الصورة (image resolution) ويعبر عن الدقة بوحدة (dot per square inch) dpi اي عدد البكسلات في الانش المربع.

وكلما كانت دقة الصورة أكبر كلما كان عدد البكسلات في واحدة المساحة أكبر، فالدقة الأكبر تعطي صوراً أكثر وضوحاً لكن لها خصائص سلبية حيث تحتاج قدرة معالجة أعلى لتحليلها وحجم ذاكرة أكبر.

فلذلك يجب الموازنة بين الموارد العتادية المتاحة لتطبيقك وبين دقة الصور الملتقطة.



الفصل الثالث

الشبكات العصبونية الالتفافية

# Convolutional Neural Networks (CNN)

CNN أو ConvNets هي فئة من الشبكات العصبونية التي تحظى بالاحترام في مجال التعرف على الصور وتصنيفها.

تستخدم CNN متعددة الطبقات والتي تتطلب التدريب لأداء مهمة التصنيف بفعالية كبيرة. تم تصميمها على غرار العمليات البيولوجية من حيث أنماط الاتصال بين الخلايا العصبية في القشرة البصرية للحيوانات. تقوم CNN بأداء أفضل من "خوارزميات التعرف على الصور والفيديو" في مجالات تصنيف الصور وتحليل الصور الطبية ومعالجة اللغة الطبيعية ولكن قبل البدء فيها من الضروري أن نتعرف على المفاهيم الأساسية في الشبكات العصبونية الصناعية Artificial Neural Network .

## 1.4 العصبون Neuron :

العصبون (neuron) : يمثل العصبون الوحدة الأساسية الصغرى لبناء الشبكة العصبونية و يقابل الخلية العصبية في النموذج البيولوجي, و هو عبارة عن عنصر حاسوبي له عدة أقنية دخل و قناة خرج واحدة, حيث تأتي أقنية الدخل بمعلومات و لكن لا تكون جميعها بنفس الأهمية و لذلك ننقل كل معلومة بوزن (weight) تكون هذه الأوزان عشوائية في البداية ثم يتم تعديلها أثناء عملية التدريب,

في بعض النماذج قد نضع لكل عصبون عتبة (base) وهي بمثابة عتبة التنبيه في النموذج البيولوجي .يجري كل عصبون العمليات التالية لإيجاد الخرج :

- جداء كل قيمة من قيم الدخل بالوزن المرفق به, ثم جمع قيم نتائج الجداء جميعها لنحصل على ما يسمى بقيمة نشاط العصبون. Neuron activity
- جمع قيمة العتبة إلى النشاط السابق.
- تطبيق تابع التفعيل (Activation function) على ناتج الجمع لإيجاد قيمة خرج العصبون.

يعتبر اختيار تابع التنشيط المناسب خطوة هامة في تصميم الشبكة العصبونية, ويكون هدفه الأساسي سحب الخرج للمنطقة المقبولة وتكون قيمها إما

[ -1,1] :إن كان bipolar ثنائي القطب أو [0,1] إن كان binary ثنائي .

## 2.4 الشبكات العصبونية متعددة الطبقات :

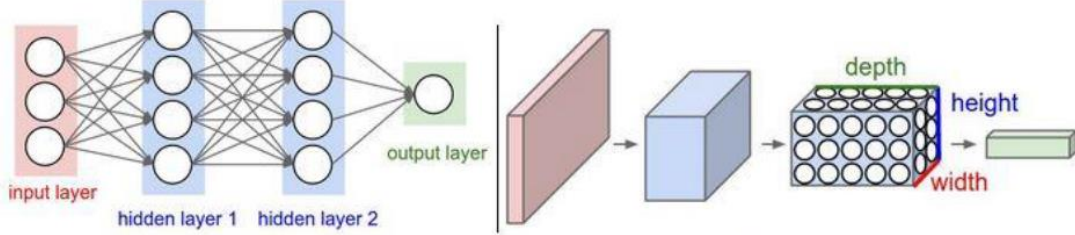
تتركب الشبكة العصبونية من مجموعة من العصبونات المرتبة ضمن طبقات تتألف بشكل أساسي من طبقة دخل و طبقة خرج و واحدة أو أكثر من الطبقات المخفية .

تعد الشبكات العصبونية بالدخل الأمامي من أهم أنواع الشبكات العصبونية متعددة الطبقات , وسميت بذلك لأنها تعتمد مبدأ الانتشار الأمامي حيث يكون خرج جميع عصبونات الطبقة  $m$  دخلاً لكل عصبون في الطبقة الموجودة أمامها .  $m+1$

## 3.4 التعلم و التدريب :

تشكل مرحلة تدريب الشبكة العصبونية مرحلة أساسية و هامة تأتي بعد مرحلة تصميمها , حيث يجري تدريبها على مجموعة من الأمثلة لتعطي الشبكة نتائج صحيحة على كل الأمثلة التي قمنا بتدريبها عليها , ولكن عملية التدريب لا تتوقف عند هذا الهدف المحدود بل يأتي امتياز فكرة الشبكات العصبونية بعملية/التعميم أي قدرة الشبكة على إيجاد المخرج الصحيح لأمثلة جديدة لم يسبق لها التدريب عليها . الخوارزمية الأكثر شيوعاً لتدريب الشبكات العصبونية هي الانتشار نحو الخلف Back propagation و هي عملية تكرارية تبدأ من الطبقة الأخيرة و تحسب الخطأ الحاصل على الخرج عبر إيجاد الفرق الحاصل بين قيمته و قيمة الخرج المطلوب , ثم تحاول تخفيض قيمة هذا الخطأ باستخدام خوارزمية تخفيض الميل العشوائي Stochastic gradient descent أو خوارزميات استمثال أخرى و التي تقوم بتعديل قيم أوزان طبقة الخرج بشكل يخفف من الخطأ الحاصل , ثم تكرر هذه العملية على الطبقة قبل الأخيرة و هكذا باتجاه الوراء وصولاً للطبقة الأولى , ثم قد تكرر جميع ما سبق على جميع الطبقات وصولاً للأوزان المثلى التي تحقق القيمة الأدنى المقبولة لخطأ الشبكة .

## 5.4 بنية الشبكة العصبونية الالتفافية :

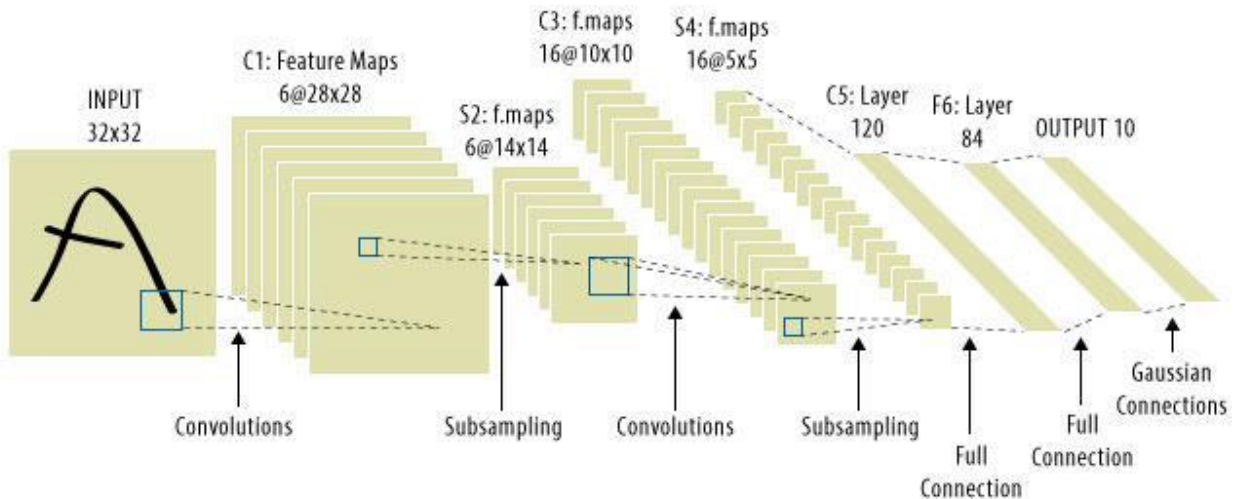


Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

الشكل (13): معمارية الشبكة العصبونية الالتفافية

كانت LeNet واحدة من أوائل CNN التي تم تعميمها وتمهيد الطريق أمام الأبحاث المستقبلية .

تم تطوير LeNet بشكل أساسي لمهام التعرف على الأحرف مثل الأرقام والرموز البريدية. منذ ذلك الحين تم إنشاء مجموعة البيانات MNIST وما زالت قيد الإعداد كمعيار لاختبار هندسة كل شبكة عصبونية جديدة مقترحة للتأكد من دقتها .



الشكل(13): شبكة عصبونية التلافية بسيطة

تأخذ شبكة CNN في الشكل أعلاه صورة بحجم  $32 \times 32$  من مجموعة بيانات الصور المكتوبة بخط اليد في MNIST وتصنفها إلى 26 حرفاً في الأبجدية الإنجليزية. هناك أربع عمليات رئيسية في الصورة أعلاه:

1.Convolution

2. غير الخطية (ReLU)

3.Pooling or Sub Sampling

4. التصنيف Classification

شرح كيفية عمل الكتل الأساسية لشبكة CNN :

### • Convolution:

تأتي تسمية هذا النوع من الطبقات من عملية الطي أو الالتفاف الرياضية وهي عملية تطبق على تابعين  $f, g$  و تنتج التابع  $o$  وهو نسخة معدلة عن التابعين ويعطي المنطقة المتداخلة بينهما , لهذه العملية أثر كبير في معالجة الصور تنتج إذاً هذه الطبقات عن تطبيق عملية الطي أو الالتفاف الرياضية على عناصر الدخل (عصبونات الطبقة السابقة أو بكسلات الصورة المدخلة ) مع عدد من المرشحات filters أو نوى الالتفاف Convolution kernels وتكرر عملية الالتفاف السابقة على كل مجموعة عناصر من الدخل بحجم المرشح الواحد, لتنتج لدينا قيم واحدة من خرائط الميزات أو , feature map ثم نكرر كامل العمليات السابقة من أجل كل مرشح فينتج خريطة ميزات أخرى وهكذا...

تعتبر قيم هذه المرشحات عن أوزان الشبكة weights نلاحظ ما يلي في هذه الطبقات :

○ الأوزان المشتركة shared weights :

تكرر تطبيق عملية التلاف المرشح على كامل عناصر الدخل سيحسن الأداء بشكل كبير لأنه سيؤدي إلى وجود عدد كبير من الوصلات و لكن بأوزان

مشاركة فيما بينها shared weights وهذه إحدى أهم ميزات هذا النوع من الشبكات لأنها تزيد فاعلية التعلم و تجعل الشبكة مؤهلة لتحقيق نتائج أفضل في التعميم, فمثلاً سيتصل كل عصبون ناتج في خريطة الميزات بوصلات بعدد عناصر المرشح ولكن أوزان هذه الوصلات ستكون ذاتها أوزان العصبون المجاور له في نفس خريطة الميزات , كما ستؤمن هذه العملية التعرف على ميزات الصورة بالنظر إلى موقعها في الصورة الأساسية لأنها تربط النقاط المتجاورة مع بعضها وتسمح أيضاً بتمثيل المناطق المتداخلة .

#### ○ مضاعفة المرشحات

وجود عدة مرشحات تطبق عليها عملية الطي مع نفس الدخل سيسمح بالتعرف على مجموعات مختلفة من الميزات في الصورة بما أن كل منها سينتج خريطة ميزات فريدة , ولكن عددها يجب أن يوضع بشكل لا يتغاضى عن تعقيد العمليات الحسابية الحاصلة كما أنه يعتمد على عدد الأمثلة المتوفرة للتدريب و تعقيد المهمة و تباين وأبعاد الصورة المدخلة, بينما يعتمد حجم المرشح الواحد أي أبعاده بشكل أساسي على البيانات الموجودة في مجموعة أمثلة التدريب.

سنأخذ مثال أرقام MNIST المكتوبة بخط اليد. إذا قمنا بتمثيل كل صورة كمصفوفة قيم pixels، فلدينا مصفوفة ثنائية الأبعاد بقيم تتراوح من 0 إلى 255.

تستمد ConvNets كلمة الالتفاف من هذه الخطوة. الغرض من الالتفاف هو استخراج الميزات feature من بيانات الإدخال سواء كانت صورة أو فيديو أو بيانات متسلسلة.

يعمل الالتفاف عن طريق الحفاظ على العلاقات المكانية بين pixels من خلال تعلم ميزات الصور feature. إذا عدنا إلى النظر في كل صورة كمصفوفة من pixels ، يقوم الالتفاف بتحويل كتل البيانات من المصفوفة إلى بعد أصغر.

|                 |                 |                 |   |   |
|-----------------|-----------------|-----------------|---|---|
| 1 <sub>x1</sub> | 1 <sub>x0</sub> | 1 <sub>x1</sub> | 0 | 0 |
| 0 <sub>x0</sub> | 1 <sub>x1</sub> | 1 <sub>x0</sub> | 1 | 0 |
| 0 <sub>x1</sub> | 0 <sub>x0</sub> | 1 <sub>x1</sub> | 1 | 1 |
| 0               | 0               | 1               | 1 | 0 |
| 0               | 1               | 1               | 0 | 0 |

Image

|   |  |  |
|---|--|--|
| 4 |  |  |
|   |  |  |
|   |  |  |

Convolved  
Feature

الشكل (15): عملية الالتفاف Convolution operation

في مصطلحات CNN ، تسمى المصفوفة  $3 \times 3$  مرشح "filter" أو "feature detector" وتشكل المصفوفة عن طريق تحريك filter فوق الصورة وتسمى بـ "feature map".

تتكرر العملية حتى يتم تحويل صورة الدخل إلى سلسلة من feature map. هناك العديد من الخيارات التي يمكن أن تستخدمها وظيفة الالتفاف convolution لإنشاء feature map مثل:

1. Edge Detection

2. Sharpen

3. Blur

يتم تحقيق كل هذه الأمور فقط عن طريق تغيير القيم الرقمية لمصفوفة المرشح قبل عملية الالتفاف ، وهذا يعني أن المرشحات المختلفة تحقق نتائج مختلفة حسب الهدف النهائي للنموذج.

من الناحية النظرية ، فإن CNN "تتعلم" قيم المرشحات أثناء عملية التدريب .

يمكن تعديل حجم المرشح والبنية وعدد المرشحات لتحقيق نتائج مختلفة.

حجم feature map هو نتاج:

1. العمق Depth: يحدد عدد الخلايا العصبونية المتصلة في الطبقة وحجم المدخلات. نقوم بتدريب هذه الخلايا العصبونية لمعرفة الميزات feature , حيث تأخذ طبقة الإدخال صورة وتستخرج الطبقات التالية الميزات من الصورة مثل الحواف والنقط وما إلى ذلك.

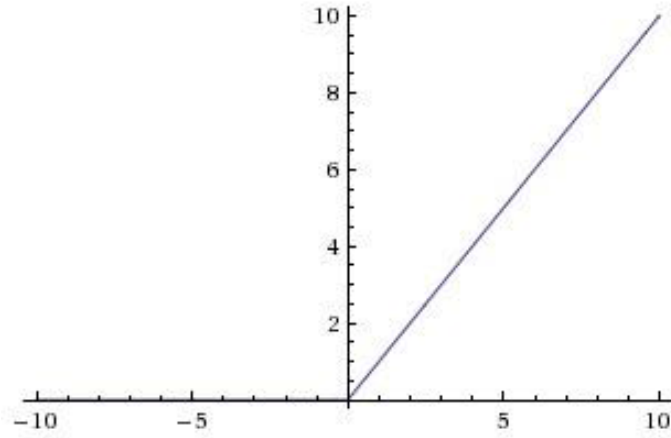
2. Stride: يشير Stride إلى عدد pixel الذي يتحرك به المرشح عبر صورة الدخل. عندما تكون الخطوة 4 ، فإنها تتحرك 4 pixel بعد تشكيل feature map. وجود خطوة أكبر تؤدي لتشكيل feature map أصغر.

3. Zero-padding : في بعض الأحيان تكون مصفوفة الإدخال محشوة بالأصفار لتطبيق المرشح على العناصر الموجودة في حدود الصورة. تسمى wide convolution الذي يختلف عن narrow convolution الذي هو الحال بالنسبة للصور غير المحشوة .

## • ReLU:

يتم استخدام العملية غير الخطية بعد عملية الالتفاف المذكورة أعلاه. ReLU تعني Unit Rectified Linear يتم تطبيقه على كل عنصر على حدة ، ويستبدل كل قيم pixel السالبة في feature map إلى الصفر. والغرض من ReLU هو تقديم غير الخطية لأن التدريب في العالم الحقيقي هو غير خطي .





الشكل (16): تابع ReLU

يعمل تابع ReLU على إعطاء حد أقصى بين رقم الإدخال و 0.

$$(x) = \max[0, x]$$

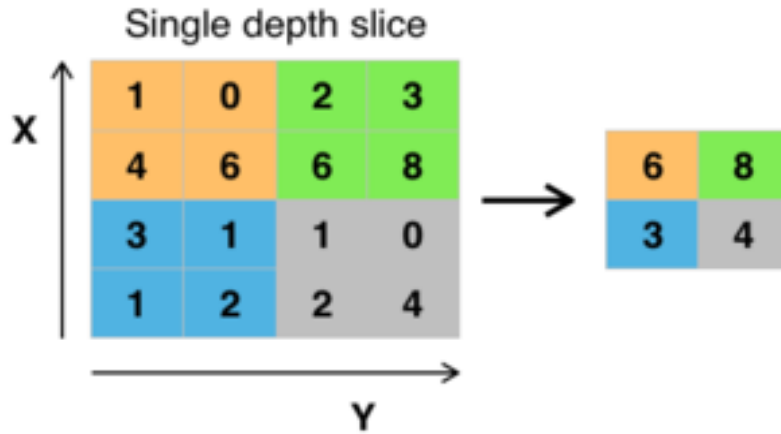
ReLU هو تابع تفعيل رخيص حسابياً على عكس توابع التفعيل الأخرى مثل sigmoid , tanh لأنها تتطلب عمليات رياضية أبسط وهي نقطة جيدة يجب مراعاتها عند تصميم الشبكات العصبونية.

## • The Pooling Step

تعد هذه الطبقات خيارية في تصميم الشبكة أي لا يشترط وجودها, وإن وجدت فستأتي بعد كل طبقة من الطبقات الالتفافية و تهدف إلى تخفيض عدد العينات أو العصبونات حيث ستقوم باختصار كل مجموعة من عصبونات الدخل بحجم معين إلى عصبون واحد, ويحدد هذا الحجم ضمن تصميم الشبكة وتكون قيمته المثلى  $2 \times 2$  لأن تكبيرها قد يؤدي إلى ضياع في المعلومات , و يتم التخفيض بعدة طرق منها:

Max Pooling : عندما نحدد نافذة بحجم معين ونأخذ أكبر عنصر منها.

بدلاً من أخذ العنصر الأكبر ، يمكننا أيضاً أخذ متوسط (متوسط التجميع) أو جمع كل العناصر الموجودة فيه (مجموع التجميع). نستمر في تحريك المرشح على الصورة .



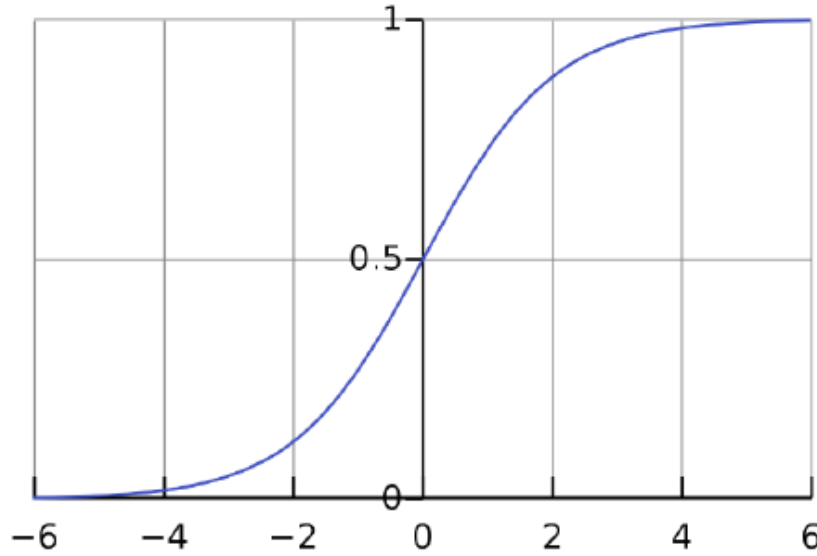
الشكل (17): طبقة Maxpooling

تقلل طبقة التجميع أيضًا من بُعد صورة الدخل وبالتالي تقلل من عدد العمليات الحسابية في الشبكة. أي أن هذه الطبقة تعطينا تمثيل لصورة الدخل بشكل أكثر إيجازًا.

### • Fully Connected Layer:

هي عبارة عن perceptron متعددة الطبقات تستخدم توابع التنشيط مثل SoftMax في طبقة الخرج. يشير مصطلح الطبقة المتصلة بالكامل إلى أن كل خلية عصبونية في الطبقة متصلة بكل خلية عصبونية في الطبقة السابقة.

تنشئ طبقة convolution جنبًا إلى جنب مع طبقة pooling تلخيصًا لصورة الدخل الأصلية التي يتم إدخالها في fully connected layer لكي تعطي خرجًا يمكن أن يكون تصنيفًا classification.



الشكل (18): تابع Softmax

تسمح هذه الطبقة بعمليات مثل back propagation وهي ميزات أساسية تمكن الشبكة العصبونية من إجراء تصنيف بدقة عالية.

## The Complete Architecture :

يمكن تلخيص تدريب CNN على النحو التالي :

1. تهيئة جميع المرشحات و parameters المحددات لأداء خطوة الالتفاف convolution على الصور المدخلة.
2. تأخذ الشبكة صورة دخل وتتصفح جميع الخطوات المذكورة أعلاه بترتيب تسلسلي وتجد ناتجاً. يتم نشر المخرجات للخلف عبر الشبكة "لتدريب" الشبكة.
3. يتم تكرار الخطوة 2 حتى تكون النواتج المتوقعة قريبة من الحقيقة ولا يمكن تغييرها.

الخطوات المذكورة أعلاه أساساً لتدريب الشبكة العصبونية لأداء مهمة محددة. بعد التدريب عندما يتم تقديم صورة جديدة إلى الشبكة العصبونية ، يمكنها التنبؤ بفئة من

الصور بناءً على مجموعة البيانات التدريبية. تلعب مجموعة البيانات التدريبية دورًا كبيرًا في دقة وأداء النموذج.

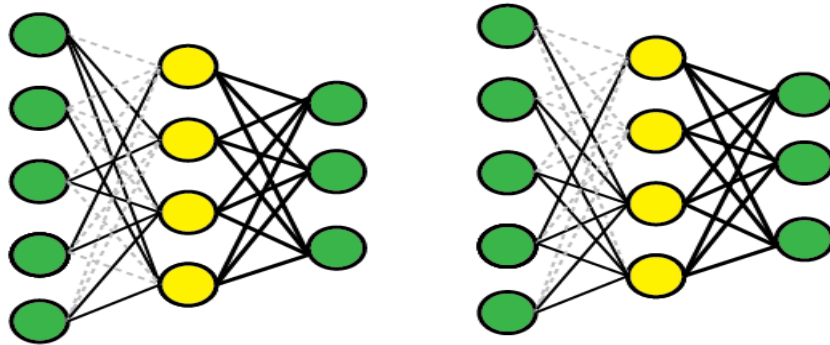
## Regularization Methods:

Regularization هو إجراء تستخدم لمنع overfitting .

1. Dropout: الطبقة المتصلة بالكامل the fully connected layer عرضة ل overfitting بسبب عدد parameters التي تشغلها . لذلك نستخدم Dropout. في كل مرحلة تدريب ، يتم "إسقاط" العصونات من الشبكة مع إما الاحتمال  $p-1$  أو الاحتفاظ بها مع الاحتمال  $p$ . يتم بعد ذلك إعادة إدخال العصونات التي تمت إزالتها في الشبكة باستخدام أوزانها الأصلية .

يقلل Dropout من overfitting عن طريق تجنب تدريب جميع العصونات على جميع بيانات التدريب. وهو يحسن بشكل كبير أيضا سرعة التدريب.

2. DropConnect : هو تعميم dropout الذي يمكن فيه إسقاط كل وصلة باحتمال  $p-1$ . لكن الفرق هنا هو جعل الأوزان مساوية للصفر وليس الخرج .



الشكل (19): شرح طبقة Dropout

## 5.3 – لمحة عن مكتبة TensorFlow :

تعتبر مكتبة TensorFlow التابعة لشركة Google من أشهر مكتبات التعلم العميق Deep Learning في الوقت الحالي. تستخدم Google التعلم الآلي Machine Learning في جميع منتجاتها لتحسين أداء محرك البحث أو الترجمة أو التسميات التوضيحية للصور.

يمكن استخدام TensorFlow لبناء أي بنية تعليمية عميقة, مثل CNN أو RNN أو ANN.

قام فريق TensorFlow التابع لـ Google Brain بملء الفجوة بين الباحثين والمطورين. ففي عام 2015, حيث جعلوا TensorFlow متاح للجميع.

تم تصميم TensorFlow ليعمل على العديد من وحدات المعالجة المركزية (CPU) أو وحدات المعالجة الرسومية (GPU) وحتى أنظمة تشغيل الهواتف المحمولة, ومتوفرة أيضا لعدة لغات مثل Python أو C++ أو Java.



## تاريخ TensorFlow :

بدأ التعلم العميق مؤخراً بالتفوق على جميع خوارزميات تعلم الآلة الأخرى "عند توافر كمية هائلة من البيانات بالطبع". مما دفع Google لاستخدام هذا التفوق لتحسين الشبكات العصبية العميقة لكل من:

• Gmail.

• محرك بحث جوجل.

تم الإعلان عن TensorFlow لأول مرة في أواخر عام 2015, بينما ظهر الإصدار الأول في عام 2017, وهو مفتوح المصدر بموجب ترخيص Apache Open Source. حيث يمكنك استخدامه وتعديله وإعادة توزيع الإصدار المعدل وبيعه "أي الإصدار المعدل" دون دفع أي شيء إلى Google.

## أجزاء TensorFlow:

يعمل TensorFlow على ثلاثة أجزاء:

• المعالجة المسبقة للبيانات.

• بناء النموذج.

• تدريب واجراء التنبؤات بالنموذج.

يطلق عليه TensorFlow لأنه يستقبل المدخلات كمجموعة متعددة الأبعاد, والمعروفة أيضاً باسم, tensors حيث تدخل المدخلات tensors من ناحية, ثم تتدفق خلال النظام عبر عمليات متعددة وتخرج من ناحية أخرى كمخرجات "أي ناتج".

-متطلبات تشغيل TensorFlow تختلف حسب الغرض منها, ويمكن تصنيفها لغرضين:

1. مرحلة التطوير: عند تدريب النموذج.

2. مرحلة التشغيل : بمجرد الانتهاء من التدريب , وأستخدام النموذج للقيام بالتنبؤات.

تستطيع تشغيل **TensorFlow** على كل من:

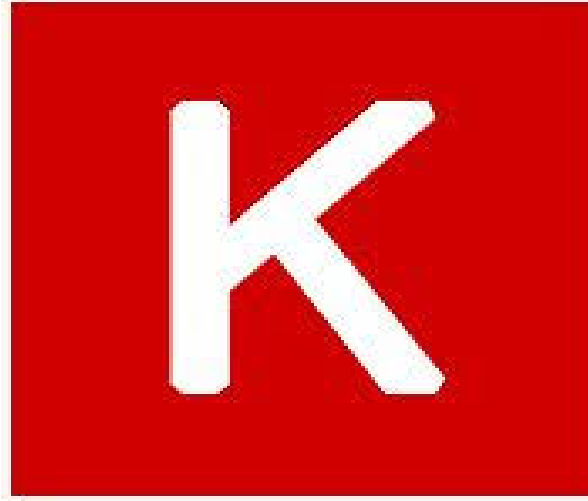
- حاسوب مكتبي يعمل بنظام ويندوز أو ماك أو لينكس.
- الحوسبة السحابية كخدمة على شبكة الإنترنت.
- أجهزة محمولة مثل iOS و Android.

يمكنك القيام بعملية التدريب على أجهزة مختلفة, وبمجرد حصولك على النموذج المدرب يمكنك تشغيله على جهاز مختلف أيضاً. يمكنك تدريب النموذج واستخدامه على كلاً من وحدات معالجة الرسوم (GPU) ووحدات المعالجة المركزية (CPU) بالطبع.

تُستعمل وحدات معالجة الرسومات (GPU) لمعالجة بيانات ألعاب الفيديو, ولكن في أواخر عام 2010 وجد باحثون في جامعة ستانفورد أن وحدة معالجة الرسومية (GPU) هذه جيدة جداً في العمليات الحسابية وعمليات الجبر و بالأخص على المصفوفة, حيث انها سريعة جداً في القيام بهذا النوع من الحسابات. يحتوي التعلم العميق على الكثير من العمليات الحسابية على المصفوفات. و TensorFlow سريع جداً في القيام بهذه العمليات لأنه مكتوب ب C++ و كميزة إضافية, يمكنك استعمال TensorFlow والتحكم فيه من قبل لغات أخرى كال Python.

ميزة أخرى مهمة في TensorFlow هي TensorBoard تمكنك لوحة TensorBoard من مراقبة ما يحدث أثناء التدريب وبشكل مرئي.

## 6.3 – لمحة عن مكتبة Keras :



مكتبة Keras هي مكتبة شبكة عصبية مفتوحة المصدر مكتوبة في Python. وهو قادر على تشغيل برنامج MXNet، التعلم العميق TensorFlow، CNTK، أو Theano. تم تصميمه ليسمح بتجربة سريعة للشبكات العصبية العميقة، ويركز على أن يكون نموذجيا وقابلا للتوسعة. تقدم هذه الدورة دليلا شاملا حول التعلم العميق باستخدام مكتبة Keras .

في عام 2017 قرر فريق Google TensorFlow دعم Keras في مكتبة TensorFlow الأساسية.

تحتوي Keras على العديد من التطبيقات لبناء الشبكة العصبية الشائعة الاستخدام مثل الطبقات وتوابع التفعيل ومجموعة من الأدوات لجعل العمل مع بيانات الصورة والنص أسهل.



الفصل الرابع

الشبكات العصبونية العودية

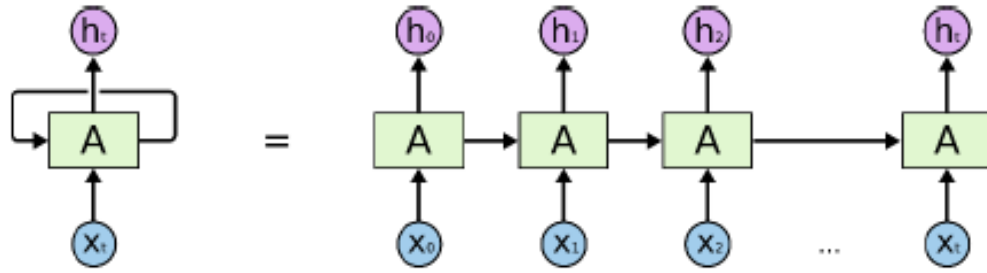
# Recurrent Neural Networks

## (RNN)

## 1.4-مقدمة:

الشبكات العصبونية المتكررة (RNN) هي فئة من الشبكات العصبونية حيث ترتبط العصبونات مباشرة لتكوين رسم بياني موجه. واحدة من ميزات RNN هي القدرة على التعامل مع التسلسل الزمني. تُعرف شبكة RNN بالدقة العالية في تسلسل البيانات وفق الزمن بسبب بنية الشبكة وتخطيطها.

تعمل RNN التقليدية على افتراض أن الخرج الناتج من طبقات ما قبل التنفيذ مهمة للدقة المتعاقبة. يمكن لـ RNN الاحتفاظ بالمعلومات التي تجعلها فعالة للغاية في التعامل مع البيانات المتسلسلة. يمكنهم الاحتفاظ بالمعلومات بسبب "الذاكرة" التي تلتقط المعلومات حول الخرج



الشكل(20): تمثيل بسيط لشبكة RNN

يوضح الشكل العمل الداخلي لشبكة RNN, حيث أنه يتم تغذية الخرج  $h$  من الطبقة السابقة إلى الطبقة التالية مع إدخال جديد  $x$  من التسلسل. تشير الأسهم إلى الذاكرة المخفية و "تخزين" المخرجات السابقة والتغذية إلى الطبقة التالية.

الحالة المخفية لـ RNN هي في الأساس ذاكرة الشبكة وهي تلتقط المعلومات من خطوات زمنية سابقة. يستخدم RNN أيضًا نفس parameters في جميع الطبقات بخلاف بنية CNN. يمكن أن تعدل وتيرة المخرجات من RNN اعتمادًا على عدد المرات المطلوبة.

## 2.4 استخدامات RNN :

يتم استخدام RNN في مهام معالجة اللغة الطبيعية في كثير من الأحيان . يُطلق على أكثر إصدارات RNN شيوعاً Long Short-Term Memory أو LSTM .  
يمكن لـ RNN أداء مجموعة متنوعة من المهام مثل:

1. نمذجة اللغة وتوليد النص: يمكن استخدام RNN في التنبؤ باحتمال حدوث كلمة في جملة وتسمى هذه النماذج "نماذج اللغة" وهي تأخذ مدخلات من الكلمات وتعطي تجميع ممكن من مخرجات الكلمات.
2. الترجمة الآلية: الترجمة الآلية مثل نمذجة اللغة التي نخرج بها سلسلة من الكلمات ، والفرق بينهما هو أن الترجمة اللغوية لا تتطلب نصاً مكتملاً لإنشاء نص جديد بينما تتطلب الترجمة الآلية نصاً مكتملاً لإعطاء مخرجات.
3. التعرف على الكلام: إذا أعطيت RNN سلسلة من الإشارات الصوتية ، فيمكن أن يتم تدريبها على التعرف على الكلام وجمل ذات معنى من الصوت المسجل.

## 3.4 مشكلة التلاشي بالتدرّج:

### Vanishing Gradient Problem:

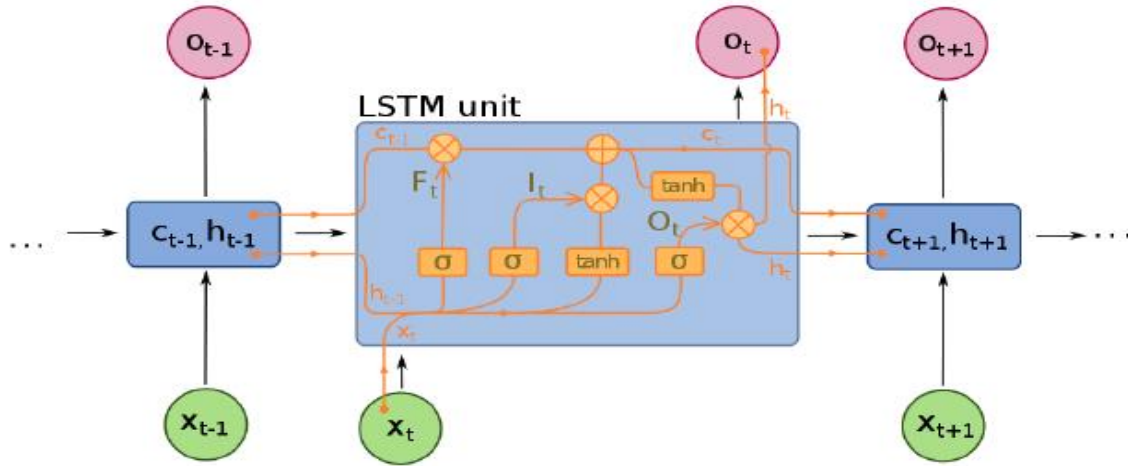
هي مشكلة موجودة في تدريب الشبكات العصبونية. في مثل هذه الشبكات ، يتم تحديث الأوزان بعد كل تمريرة. ولكن المشكلة هي في بعض الأحيان أن التدرج سيكون صغيراً ، مما يمنع الوزن من تغيير القيم. أسوأ الحالات هي أن الشبكة العصبونية تتوقف عن التحديث بالكامل.

لنأخذ مثالا على وظيفة التفعيل التقليدية مثل وظيفة tanh التي لها تدرجات في النطاق (0 ، 1) وتستخدم backpropagation. على الطبقات n ، يتناقص التدرج بشكل كبير وتندرج الطبقات الأمامية n ببطء شديد .  
حلول المشكلة السابقة :

1. Multi-level hierarchy: اقترحه Jurgen Schmidhuber للشبكات التي يتم اختبارها مسبقا من خلال التعلم غير الخاضع للإشراف unsupervised learning ومن ثم ضبطها من خلال backpropagation.
2. LSTM: سيتم تغطيتها في القسم التالي.
3. Faster hardware : قد تم تطوير العتاديات منذ تم إدخال الشبكات العصبونية إلى عالم الحوسبة ، مما يجعل backpropagation ممكنا لعدة طبقات عميقة .
4. وظائف التنشيط الأخرى: المحولات مثل ReLU تعاني أقل من مشكلة Vanishing Gradient.

## **LSTM:**

تم اكتشاف شبكات Long short-term memory (LSTM) في عام 1997 وقد حققت نتائج دقيقة في مجالات متعددة. عام 2007 ، بدأ استخدام LSTM على نطاق واسع للتعرف على الكلام بنتائج لا تصدق و في عام 2009 ، فازت بنية RNN القائمة على التعرف على الأنماط بالعديد من المسابقات من خلال تحديد معايير للدقة في التعرف على البيانات المكتوبة بخط اليد.



الشكل (21): بنية LSTM

LSTM هي بنية للتعلم العميق مصممة لتجنب مشكلة Vanishing Gradient التي تواجهها نماذج التعلم العميق أثناء التدريب والتي تنشأ بسبب فقدان الدقة أثناء backpropagation.

يسمح backpropagation لكل من أوزان الشبكات بتحديث قليل بناءً على تقدم التدريب ، ولكن في بعض الأحيان يكون التدرج صغيراً جداً بحيث أنه يمنع الأوزان من تغيير القيم التي قد تؤدي إلى توقف الشبكة العصبونية بالكامل عن التدريب.

لذلك قد تم استخدام عدة حلول لهذه المشكلة ومنها LSTM .

يتم تعزيز LSTM من خلال البوابات المتكررة التي تسمى بوابات "forget" والذي يمنح LSTM إمكانية تعلم المهام التي تتطلب ذواكر للأحداث التي حدثت ملايين من المرات في وقت سابق.

LSTM محددة للغاية ويمكن تعديلها لمهام معينة عند الحاجة إليها.

## Training an RNN:

إن الخطوات المطلوبة لتدريب شبكة RNN تشبه إلى حد بعيد تدريب شبكة CNN كما ذكرنا في الفصل السابق. مفهوم backpropagation هو نفسه ، إلا أن هناك تعديلات طفيفة على العملية نفسها.

بخلاف CNN ، لا تتغير parameters بين الطبقات في RNN ، وبالتالي تتم مشاركتها بواسطة جميع الخطوات الزمنية في الشبكة ، لا يعتمد الخرج على العمليات الحسابية المطبقة على الخطوة الزمنية الحالية وحسب بل يعتمد على الخطوات الزمنية السابقة أيضا .

يتطلب تدريب RNN في خطوة زمنية معينة حساب التدرجات  $n$  وراءها. تسمى هذه العملية (BPTT) Backpropagation Through Time ، وهو المبدأ الأساسي وراء بنية RNN.

من الناحية التقليدية ، يعاني RNN من مشكلة vanishing gradient ، ولكن توجد RNN المتخصصة (LSTM) والتي لا تعاني من هذه المشكلة.

# الفصل الخامس

## مقدمة في الأندرويد

# Android





## الشكل (22): شعار نظام التشغيل Android

Android : هو أساساً نظام تشغيل مفتوح المصدر صمم للأجهزة ذات شاشات اللمس كالهواتف الذكية والحواسب اللوحية , مبني على نواة لينوكس, يدعم تعدد التطبيقات , أي من الممكن أن يعمل به أكثر من تطبيق في نفس الوقت أو الخدمات في الخلفية، وتعمل كل التطبيقات فيه بشكل متساوي من ناحية مشاركة موارد الجهاز.

وهو مجاني بحيث أن أي شخص يمكن أن يقوم بالبرمجة فيه والتعديل عليه، كما أن لغة جافا تستعمل واجهات مرئية (XML) سهلة الاستخدام لذا ستجعلنا نقوم بعمل برامج Android بسهولة.

## 1.5 تاريخ نظام Android:

تأسس نظام Android في بالو ألتو كاليفورنيا بالولايات المتحدة الأمريكية والأعضاء الرئيسيين المشاركين بتأسيس نظام Android هم Andy Rubin و Nick Sear و Chris White.

كانت الأهداف الأولية لشركة تطوير نظام معالجة متقدم للكاميرات الرقمية وعندما أدركت الشركة أن سوق أجهزة الموبايل ذو طلب متزايد توجهت لإنتاج نظام معالجة هواتف ذكية ينافس نظام Symbian و Microsoft.

طور فريق من شركة Google تحت إشراف Andy Rubin منصة لجهاز الخليوي مدعومة بنواة Linux وقد سوقت غوغل هذه المنصة على وعد التزود بنظام مرن ومحدث.

واستمرت التكهّنات حول نية شركة Google بالدخول إلى سوق اتصالات الهاتف المحمول حتى عرضت التصميم الأول لجهاز Sooner وهو يشبه هاتف Blackberry ولكن دون شاشة لمس وأعيد هندسته لاحقا ليدعم شاشات اللمس وليتنافس مع الأجهزة الأخرى ك iPhone.

في الخامس من تشرين الثاني 2007 كشفت Android إنتاجها لأول منصة جهاز موبايل مبني على نواة Linux بالإصدار 2.6.25.

أطلق في 22 تشرين الأول 2008 الموبايل التجاري الأول الذي استخدم نظام Android واسمه HTC Dream.

تم تصميم نظام Android بشكل أساسي لتخديم الأجهزة ذات الشاشات اللمسية للأجهزة الذكية والأجهزة اللوحية وهو نظام تشغيل مفتوح المصدر Open source مما يسمح لمصممي الأجهزة بتكييفه ليتلاءم مع منتجاتهم ومما يزيد من قوة هذا النظام هو وجود عدد كبير من التطبيقات المتاحة مجاناً لمستخدمي نظام التشغيل هذا. تتم كتابة تطبيقات Android باستخدام لغة جافا وتتم ترجمة الكود مع البيانات المتعلقة بالبرنامج باستخدام أدوات Android SDK إلى حزمة ذات لاحقة apk .

عندما يتم تحميل التطبيق على جهاز Android يتم إنشاء Create كل تطبيق وإعطائه صلاحيات أمنية خاصة والذي يدعى Security Sandbox:

- نظام Android هو نظام Linux متعدد المستخدمين، حيث يكون كل تطبيق من التطبيقات العاملة هو مستخدم مستقل.
- يتم إعطاء كل تطبيق من التطبيقات العاملة رقما فريدا Unique ID يتم التعامل مع هذا الرقم من قبل نظام التشغيل فقط وهو غير معروف بالنسبة للتطبيق، ويقوم نظام التشغيل بإعطاء صلاحيات للملفات التابعة لتطبيق ما، تكون هذه الصلاحيات مرتبطة بالرقم المعرف ID لهذا التطبيق حيث يمكن لهذا التطبيق فقط التعامل مع الملفات.
- كل إجراء Process يتم تنفيذه على Virtual machine خاصة وبذلك يتم تنفيذ كل تطبيق بمعزل عن التطبيقات الأخرى ويقوم نظام التشغيل باستبدال تطبيق ما عندما يطلب أي من مكوناته التنفيذ execution.
- بهذه الطريقة يقوم نظام التشغيل بإعطاء أقل صلاحية ممكنة للتطبيقات، بحيث يمكن للتطبيقات الولوج فقط إلى المكونات التي يحتاجها لإتمام عمله ليس أكثر، وهذا يساعد في خلق بيئة عمل آمنة لا يمكن للتطبيق فيها الولوج إلى مصادر لم يتم إعطاؤه صلاحيات عليها.
- يمكن لتطبيقين أن يتشاركان في المصادر والبيانات نفسها من خلال إعطاء هذين التطبيقين نفس الرقم المعرف ID أو من خلال طلب أحد التطبيقات صلاحية الدخول إلى الدخول إلى مصادر وبيانات تطبيق آخر (مثل الرسائل، أرقام الهاتف...).

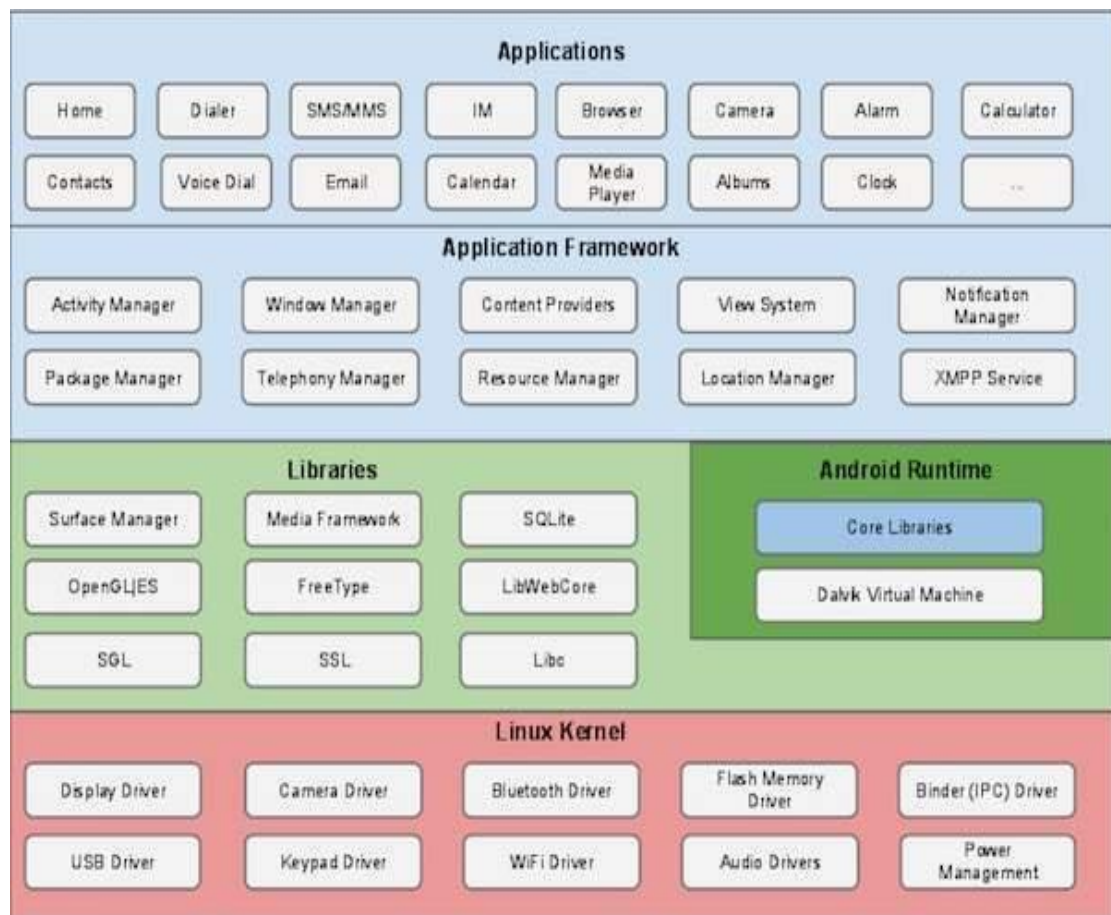
## 2.5 مميزات نظام Android:

تعتبر الأجهزة ذات نظام تشغيل Android أجهزة متكاملة حيث لم تعد مقتصرة على إجراء مكالمات هاتفية فقط بل أصبحت تدعم آخر صيحات التكنولوجيا بالإضافة إلى العديد من الميزات وهي:

- **التخزين:** يستخدم Android لتخزين البيانات SQLite أو نسخ خفيفة من قواعد البيانات المترابطة.
- **نظم الاتصال:** يدعم Android الأنظمة : Wi-Fi, Bluetooth, WiMAX, GSM.....
- **الرسائل:** يدعم خدمة الرسائل القصيرة SMS، خدمة رسائل الوسائط المتعددة MMS.
- **دعم اللغات:** يدعم Android العديد من اللغات البشرية، وتم مؤخرا دعم اللغة العربية.
- **متصفح الويب:** المتصفح الموجود على Android مبني على واجهة ( Web Kit)  
مقترن مع محرك الجافا سكربت في متصفح كروم.
- **دعم الوسائط:** يدعم النظام كل الصيغ القياسية للوسائط Media من صور وأصوات وفيديو.
- **المجانية:** يعتبر نظام Android أكثر نظام مجاني حيث أن عدد التطبيقات المجانية فيه يفوق عدد التطبيقات المدفوعة.

- أدوات الواجهة Widgets: هي تطبيقات صغيرة توجد على سطح المكتب مثل: تطبيق القدر المتبقي من طاقة البطارية، التقويم.... الخ.

## 3.5 بنية Android (Android Architecture):



الشكل (23): المكونات الأساسية لنظام أندرويد

## 4.5 بيئة Android Studio:



الشكل (24): شعار برنامج Android studio

وهو بيئة التطوير الرسمية IDE integrated development environment لأجهزة Android

تم الإعلان عنه في 2013/5/16 في مؤتمر Google I/O ، وهذه البيئة هي مجانية تماماً.

Android Studio كان في مراحل الاختبار الأولي بدءاً من الإصدار 0.1 في 2013/5، بعدها دخل في مرحلة الإصدار التجريبي beta stage بدءاً من الإصدار 0.8 التي تم إصدارها في 2014/6.

أول نسخة مستقرة من هذا البرنامج أصدرت في 2014/12، والتي كان رقمها 1.0. Android studio صمم خصيصاً لتطوير برامج Android ، وهو متاح للتنزيل مجاناً للجميع على أنظمة التشغيل Linux, MacOS, Windows, و يمكن استخدام Eclipse Android (ADT) Development Tools كبديل عنه .

## بعض المزايا :

- إعادة الاستخدام و الإصلاح السريع للأخطاء .
- Lint tools و التي تستخدم من أجل مراقبة الأداء , التوافق مع الإصدارات الأخرى , و كشف المشاكل ( حيث أن Lint tools هي مجموعة من أدوات نظام Unix تستخدم للإشارة إلى مكان الأخطاء و البرامج المشبوهة).
- دعم إمكانية بناء وتصميم تطبيقات لأجهزة القابلة للارتداء التي تعمل بنظام أندرويد Android Wear
- وجود جهاز أندرويد افتراضي، يستخدم من أجل اختبار البرنامج وكشف الأخطاء دون الحاجة لتنصيب البرنامج على جهاز حقيقي.
- يحتوى على واجهه يكون من السهل على المبرمج أن يقوم بعملية البرمجة بشكل سهل وسلس.
- Android studio تم بنائه وبرمجته على أساس يخدم برمجة تطبيقات اندرويد, بينما Eclipse هو عام وشامل ويحتاج لأدوات مشان يدعم تطوير وبرمجة التطبيقات.
- عند اضافة مكتبة برمجياً لا تحتاج سوا ادراج كود واحد فقط, مثال اضافة مكتبة إعلانات جوجل فقط سطر لتشغيلها, اي لا تحتاج تحميلها سابقاً ووضعها في مجلد Libs حال برنامج Eclipse.

## System requirements

### Version 2.x

|                   | Windows   | OS X/macOS  | Linux                |
|-------------------|---|---|----------------------|
| OS version        | Windows 10/8/7 (32-bit or 64-bit)   | Mac OS X 10.9.5 or higher, up to 10.11.6 (El Capitan) or 10.12.3 (Sierra) | GNOME or KDE desktop |
| RAM               | 4 GB RAM minimum, 8 GB RAM recommended  |   |                      |
| Disk space        | 500 MB disk space for Android Studio, at least 1.5 GB for Android SDK, emulator system images, and caches |   |                      |
| Java version      | Java Development Kit (JDK) 8  |   |                      |
| Screen resolution | 1280x800 minimum screen resolution  |   |                      |

### Version 1.x

|                       | Windows   | OS X/macOS  | Linux   |
|-----------------------|---|---|---|
| OS version            | Microsoft Windows 10/8.1/8/7/Vista /2003/XP (32 or 64 bit)        | Mac OS X 10.8.5 or higher, up to 10.10 to up 10.10.2 up 10.10.3 or 10.10.5 (Yosemite) | GNOME or KDE or Unity desktop on Ubuntu or Fedora or GNU/Linux Debian |
| RAM                   | 3 GB RAM minimum, 4 GB RAM recommended                            |   |   |
| Disk space            | 500 MB disk space   |   |   |
| Space for Android SDK | At least 1 GB for Android SDK, emulator system images, and caches |   |   |
| JDK version           | Java Development Kit (JDK) 7 or higher                            |   |   |
| Screen resolution     | 1280x800 minimum screen resolution                                |   |   |

الشكل (25): متطلبات النظام الذي يحتاجه لتشغيل Android Studio



## 6.5 مكونات التطبيق Application components:

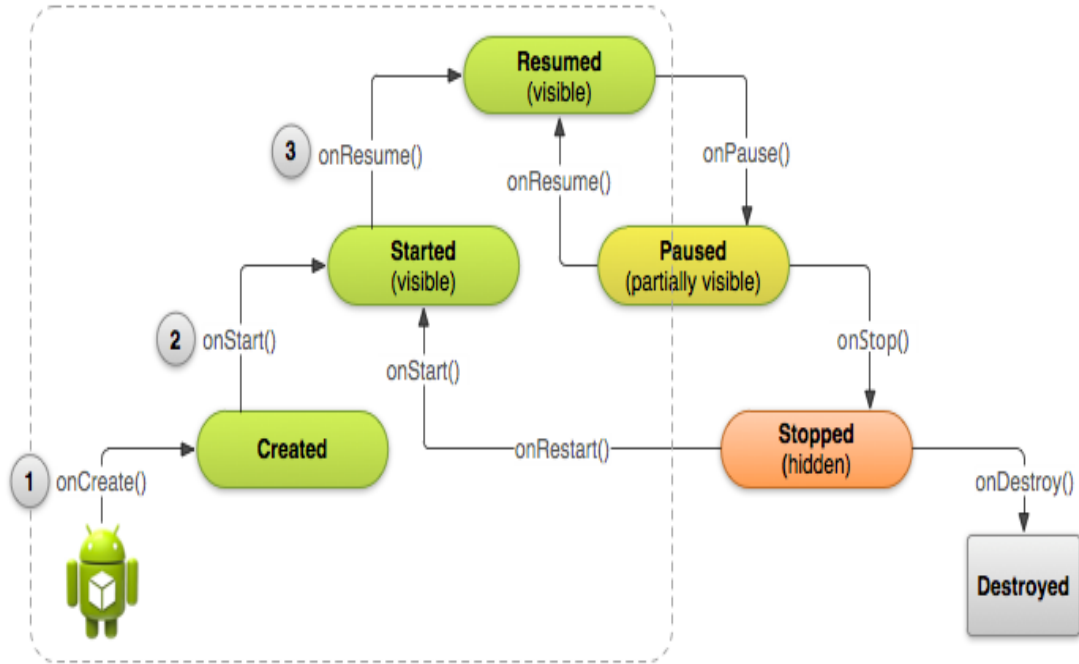
تعد مكونات التطبيقات لبنة أساسية لبناء تطبيقات الاندرويد . اذ يتم بناء كل تطبيق من خلال مزج بعض أو كل تلك المكونات ، وحيث يمكن أيضاً الاستناد إليها بشكل فردي .

هناك أربعة مكونات رئيسية للاندرويد وهي:

### ● النشاط ( Activity ) :

للقيام بنشاط ما يجب توفر شاشة تمكن المستخدمين من التفاعل والقيام بذلك النشاط. يمكن للمستخدمين إجراء عمليات مثل إجراء مكالمة و إرسال الرسائل القصيرة، وما إلى ذلك فعلى سبيل المثال: شاشة تسجيل الدخول الخاصة بتطبيق الفيسبوك, تعتبر الفعالية التي تظهر في بداية إقلاع التطبيق الفعالية الرئيسية main ومن ثم يمكن لكل فعالية استدعاء فعالية أخرى وهكذا من أجل تنفيذ المهام المختلفة وعندما تبدأ فعالية ما تنتهي أو تتوقف الفعالية السابقة ويتم وضعها في مكدس LIFO يدعى Back stack ولاحقاً عندما يضغط المستخدم على زر العودة Back يتم إغلاق الفعالية الأخيرة التي تم تشغيلها ومتابعة آخر فعالية تم وضعها في المكدس. يتم التعامل مع الفعالية كصف جزئي SubClass من الصف Activity.

ويتم فيه تعريف الطرق التي سيستدعيها النظام عن انتقال الفعالية بين المراحل المتعددة لدورة حياتها مثلاً onCreate() و onPause(). يمكن جعل الفعالية تبدأ عن طريق استدعاء التابع startActivity مع تمرير Intent توصف النشاط ويتم إنهاء الفعالية عن طريق استدعاء التابع () finish.



الشكل (26): دورة حياة الفعالية

## ● الخدمة (service) :

تتمثل بالعمليات التي يتم تشغيلها باستمرار في خلفية الهاتف دون الحاجة لواجهة المستخدم. على سبيل المثال: مشغل الموسيقى كما ويمكن استدعاء خدمة ما service من قبل مكون لتطبيق آخر غير الذي تتبع له الخدمة.

يمكن للخدمة أن تأخذ إحدى الصيغتين التاليتين:

### ○ البداء started:

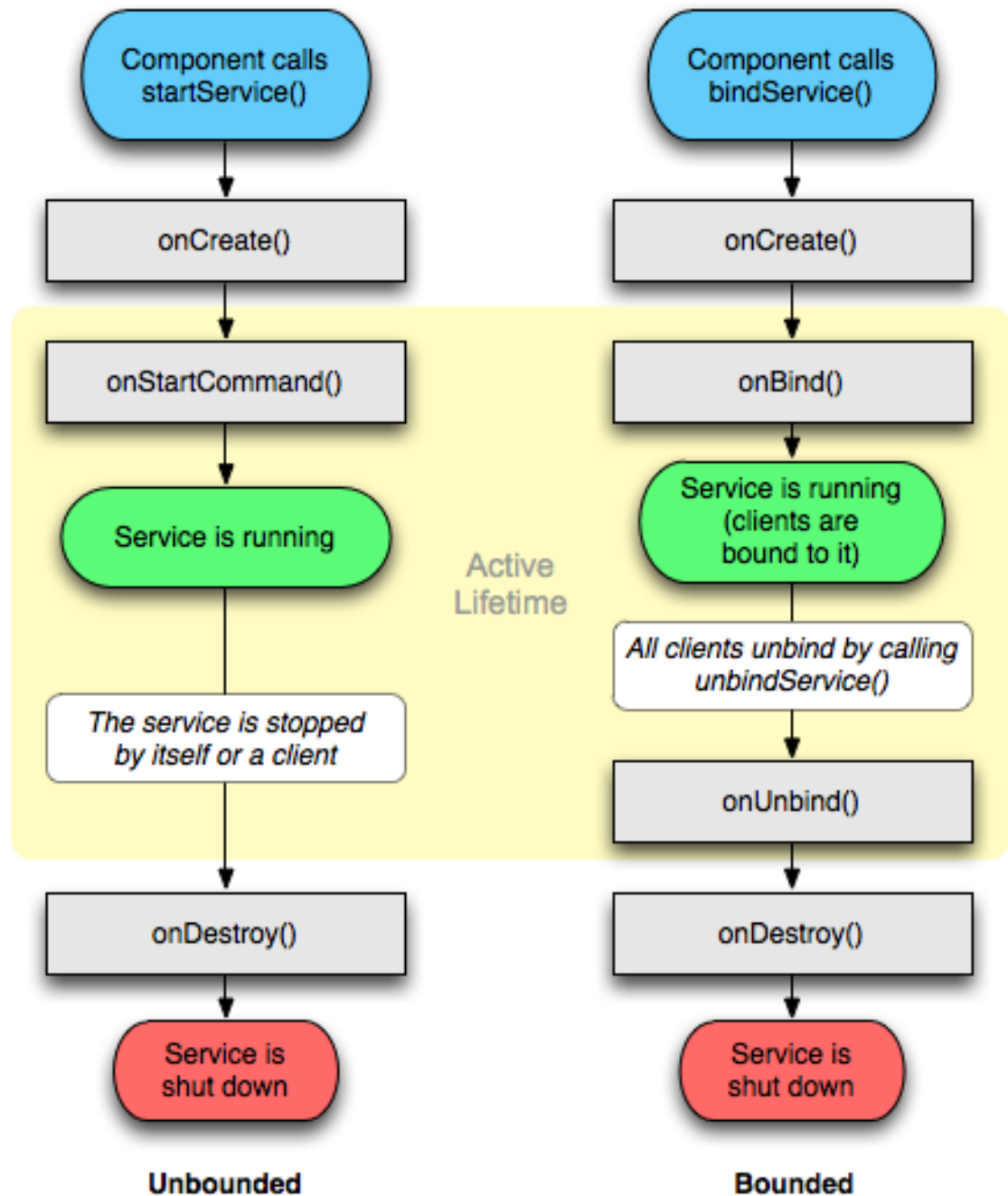
وهي الحالة التي تصبح عليها عندما يتم بدؤها من قبل فعالية ما مثلاً باستخدام `startService()` وبعدها يمكن للخدمة أن تبقى فعالة حتى بعد أن يتوقف مكون التطبيق الذي استدعاها.

### ○ حالة الارتباط Bound:

يتم ارتباط الخدمة بمكون تطبيق ما عندما يستدعي هذا المكون `bindService` والتي تسمح بتشكيل واجهة عميل-مخدم مما يمكن

مكون التطبيق من التفاعل مع هذه الخدمة، إرسال الطلبات إليها واستلام النتائج منها وتعمل الخدمة في هذه الحالة طالما أن المكون الذي طلب الارتباط بها يعمل ويمكن أن تكون مرتبطة بأكثر من مكون في وقت واحد. يتم التعامل مع الخدمات صف جزئي SubClass من الصف Service.

ويوضح الشكل التالي دورة حياة الخدمة:



الشكل (27): دورة حياة الخدمة

## ● مكونات المحتوى (content component) :

تقوم بتقديم البيانات إلى التطبيقات الخارجية على شكل جدول واحد أو أكثر. وبعبارة أخرى، يمكن اعتبار مزودات المحتوى كواجهات المستخدم التي تربط البيانات في عملية واحدة مع رمز تشغيل في عملية أخرى. على سبيل المثال: يمكن قراءة الرسائل القصيرة من أي تطبيق على الجهاز يوفر خاصية الرسائل القصيرة.

يحتوي نظام أندرويد على مزودات جاهزة لإدارة بيانات الصورة والسمعيات والفيديو وبيانات دليل الهاتف، وفي حال الحاجة إلى التعامل مع طبيعة خاصة للبيانات وتزويدها للتطبيقات الأخرى فيجب تزويد التطبيق بمزود محتويات خاصة، ويتم تقديم البيانات إلى التطبيقات الأخرى على شكل جداول شبيهة بجدول قواعد البيانات.

## ● مستقبل البث (Broadcast receivers) :

يعد جهاز استقبال البث المكون الذي يستجيب لإشعارات النظام مثل البطارية منخفضة، اكتمال التشغيل، قابس السماع الخ على الرغم من أن معظم مستقبلات البث نشأت من قبل النظام، إلا أنه يمكن للتطبيقات أيضا أن تقوم بعمليات البث.

يتمتع نظام التشغيل Android بميزة تصميمية فريدة وهي إمكانية أن يقوم تطبيق ما باستخدام أحد مكونات تطبيق آخر، وهذا ما يساعد المطور من أجل استخدام كتل أخرى مبنية سابقا بدون إنشاء كامل التطبيق بدءا من الصفر فلا لتقاط صورة مثلا، ليس على المبرمج أن يكتب نصا برمجيا يقوم بهذه العملية وإنما يمكنه ببساطة القيام ببدء الفعالية الخاصة في تطبيق الكاميرا والذي يقوم بالتقاط الصورة وعند انتهاء العملية يتم إعادة الصورة إلى التطبيق الأساسي من أجل استخدامها وبالتالي يبدو للمستخدم كما لو أن تطبيق الكاميرا هو جزء من التطبيق الذي يتعامل معه.

وعندما يقوم النظام بتشغيل مكون ما فإنه يقوم بتشغيل الإجراءات للتطبيق الذي يتبع له هذا المكون، وبدء الصفوف اللازمة لعمل هذا المكون، فعلى سبيل المثال: إذا قام تطبيق ما بتشغيل فعالية التقاط الصورة الموجودة في تطبيق الكاميرا، فإن تنفيذ هذه الفعالية يتم من خلال الإجراءات Process التابعة

لتطبيق الكاميرا، وليس في إجرائية التطبيق المستدعي لهذه الإجرائية، وهذه النقطة تميز Android عن تطبيقات الأنظمة الأخرى، حيث لا تحتوي على نقطة دخول وحيدة Single Entry Point() وبالتالي لا يوجد main() بسبب أن نظام تشغيل Android يقوم بتشغيل كل تطبيق ضمن إجرائية منفصلة لها صلاحيات محددة تمنعه من الولوج إلى تطبيقات أخرى، لذا فإن أي تطبيق لا يمكنه تفعيل مكون ما من مكونات تطبيق آخر إلا من خلال نظام التشغيل عن طريق إرسال رسالة إلى النظام تخبره بنية Intent هذا التطبيق باستخدام المكون المطلوب ليقوم النظام بتفعيل هذا المكون.

## 7.5 ملف Manifest File:

قبل بدء أي مكون من مكونات التطبيق يتعين على النظام أن يعلم بوجود هذا التطبيق عن طريق قراءة AndroidManifest.xml والذي يحتوي أيضا على:

- التعريف بصلاحيات المستخدم التي يحتاج إليها التطبيق (كقراءة محتويات دليل الهاتف مثلا).
- التعريف بال Hardware و Software اللازم من أجل عمل التطبيق
- التعريف بأقل API Level يمكن أن يعمل عليها التطبيق، وهي رقم إصدار نظام التشغيل Android version .

عند تعريف كائن Component في ملف Manifest File التابع لتطبيق ما يمكن تضمين

Intent filters تقوم بتعريف إمكانيات هذا الكائن في الاستجابة ل Intent من تطبيقات أخرى يمكن تعريف Intent filter من خلال إضافة العنصر <intent element> تحت تعريف الكائن المطلوب. بعض الخواص المطلوبة في التطبيق التي يمكن مراعاتها في ملف Manifest:

- حجم الشاشة وكثافتها:

Screen: normal, large, extra large.

Density: low, medium, high, extrahigh.

- إعدادات الإدخال:

(Hardware, Keyboard, Trackball) ومن النادر استخدامها في التطبيقات.

- ميزات الجهاز:

كأن يكون له الكاميرا مثلا، حساس إضاءة أو Bluetooth...ومن الهام جدا تعريف الإمكانيات المطلوبة من أجل عمل تطبيق ما.

الفصل السادس

بروتوكول الاتصال وقسم الإرسال

**Communication Protocol  
and Transmitter Section**



## 1.6 لمحة عن الشبكات :

الشبكة هي مجموعة من الحواسيب والأجهزة المحيطية وأجهزة الاتصالات القائمة بذاتها تتصل فيما بينها لتتشارك في الموارد والأجهزة المتصلة بالشبكة.

## 2.6 معايير تنظيم الشبكات :

كان الكيان المادي (hardware) يأخذ الاهتمام الأول في تصميم الشبكات وذلك نظراً لارتفاع تكلفته وتعقيده، ويأتي بعده الكيان البرمجي (software)، ولكن من أجل التقليل من تعقيد تصميم الشبكات قسمت الشبكات إلى طبقات أو مستويات، حيث تحقق كل طبقة مجموعة من الوظائف وتعتمد على الطبقات السفلى في إنجاز هذه الوظائف وتقديمها إلى الطبقات الأعلى.

## 3.6 البروتوكول الرسمي للإنترنت TCP/IP :

هو اختصار Transmission Control Protocol/Internet Protocol وهو معيار يضم مجموعة بروتوكولات مطورة في نهاية السبعينات من القرن الماضي من وكالة مشاريع أبحاث الدفاع المتقدمة (Defense Advanced Research Projects Agency) [DARPA] ، كطرق للتواصل بين مختلف أنواع الحواسيب وشبكات الحواسيب؛ إن بروتوكول TCP/IP هو العصب المحرك للإنترنت، وهذا ما يجعله أشهر مجموعة بروتوكولات شبكية على وجه الأرض.

يوفر هذا النموذج التناسق بين جميع أنواع بروتوكولات الشبكة وخدماتها عن طريق وصف ما ينبغي عمله في طبقة معينة، ولكن ليس عن طريق توضيح كيفية إتمام ذلك. ولا يُعنى بالنموذج المرجعي أن يكون عبارة عن مواصفات للتطبيق أو يقوم بتوفير مستوى كافٍ من التفاصيل لتحديد خدمات بنية الشبكة بدقة. والغرض الأساسي من النموذج المرجعي هو المساعدة في استيعاب أوضح للوظائف والعمليات المضمنة.

يتكون هذا المعيار من أربع طبقات :

- طبقة التطبيقات Application Layer.
- طبقة النقل Transport Layer.
- طبقة الشبكة Network Layer.
- طبقة وصل البيانات Data Link Layer

## 4.6 النموذج المرجعي OSI :

اختصار ل (Open Systems Interconnection) , ظهر لمعالجة مشكلة عدو توافق

الشبكات الناتجة عن استخدام الشركات الخاصة لمعايير و أنظمة ذاتية خاصة بشبكاتها , و نظراً للتوسع الهائل في أعداد هذه الشبكات الخاصة, قامت المنظمة الدولية لتوحيد المعايير عام 1977 (ISO International Organization for Standardization) هذه المنظمة أسستها الأمم المتحدة سنة 1946) بإنشاء طراز بيني بين الأنظمة المفتوحة يساعد مصممي الشبكات على إنشاء شبكات متوافقة مع الشبكات الأخرى .

في نموذج OSI المرجعي , هناك سبع طبقات مرقمة تقوم كل منها بتوضيح وظيفة معينة على الشبكة . يوفر تقسيم الشبكة إلى سبع طبقات الميزات التالية :

|              |
|--------------|
| Application  |
| Presentation |
| Session      |
| Transport    |
| Network      |
| Data link    |
| Physical     |

- تقسيم اتصال الشبكة إلى أجزاء أصغر قابلة للإدارة بصورة أكبر.
- توحيد معايير مكونات الشبكة لإتاحة التطوير و الدعم من قبل جهات بيع معتمدة.
- السماح لأنواع مختلفة من أجهزة و برامج الشبكة بالاتصال ببعضها البعض.
- منع تأثير التغييرات التي تحدث في إحدى الطبقات على الطبقات الأخرى.
- تقسيم اتصال الشبكة إلى أجزاء أصغر لتسهيل فهمه.

## 5.6 بروتوكول نقل النص التشعبي HTTP:

### Hypertext Transfer Protocol (HTTP)

هو مجموعة قواعد نقل الملفات (النصوص والصور الرسومية والصوت والفيديو وملفات الوسائط المتعددة الأخرى) على شبكة الويب.

تم انشاء هذا البروتوكول لتأمين نقل بيانات بين الخادم والعميل حيث يتم التواصل بينهم عن طريق الطلب والاستجابة. Request / Response

حيث عند انشاء اي طلب من العميل في صورة رسالة يتم تضمين بعض الاشياء الضرورية عن الطلب ويتم انشاء وصلة او Connection بين العميل والخادم ويستجيب الخادم للطلب مع بعض البيانات عن حالة الاستجابة من حيث النجاح او الفشل.

#### بعض الحقائق عن بروتوكول HTTP :

- تم صياغة المصطلح HTTP من قبل تيد نيلسون.
- البروتوكول HTTP مبني على رأس بروتوكولات TCP/IP وهي اساس الاتصال في الانترنت.
- بروتوكول HTTP ليس الوحيد المستخدم في الويب فهناك بروتوكولات كثيرة منها بروتوكول نقل الملف FTP.
- المنفذ القياسي المستخدم في بروتوكول HTTP هو المنفذ 80.
- HTTP/ 0.9 كان الإصدار الأول من HTTP ، وقدم في عام 1991.
- صدر HTTP / 1.0 في عام 1996.
- صدر HTTP / 1.1 رسميا في يناير 1997.

#### مراحل عمل بروتوكول HTTP :

يمكننا تقسيم عمل بروتوكول HTTP إلى أربع مراحل كالتالي:

- إنشاء الوصلة او Connection مع الخادم او Server.
- إرسال الطلب إلى الخادم.

- استلام الاستجابة من الخادم.
- إغلاق الوصلة.

\

## 6.6 بروتوكول التحكم بالنقل TCP:

### Transmission Control Protocol (TCP)

يقوم هذا البروتوكول بتوفير خدمات تعتمد على الاتصال بين أجهزة الشبكة حيث لا تتم عملية تبادل البيانات بين الأجهزة إلا إذا كان هناك اتصال مسبق بينها ويمكننا تلخيص طريقة عمل البروتوكول TCP بالخطوات الأساسية التالية:

1- إنشاء اتصال (مجرى للبيانات) بين البرنامج المرسل والبرنامج المستقبل.

2- تبادل البيانات عن طريق هذا المجرى.

3- إغلاق المجرى.

حيث يقوم TCP بالمهام التالية:

1- تجزئة وتجميع البيانات.

2- الإشعار بالاستلام Acknowledgment .

3- تحديد منافذ عبور البيانات Ports.

4- كشف الأخطاء Error Detection.

5- التحكم في تدفق البيانات.

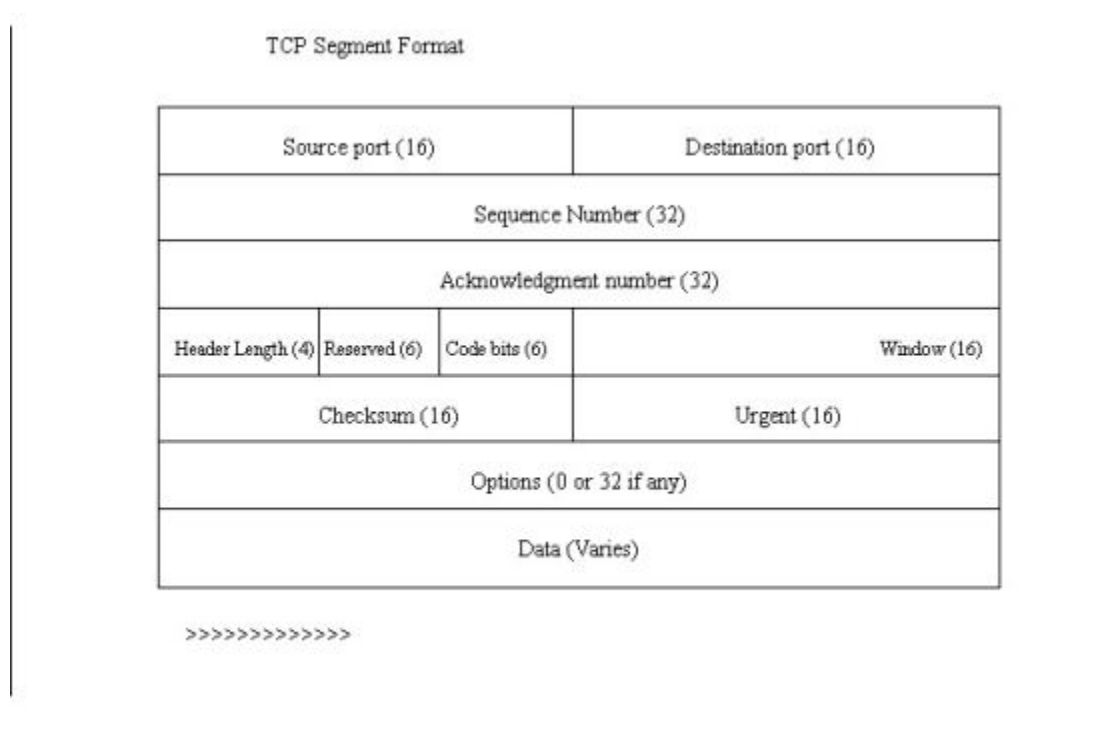
6- ترقيم الرزم Packets Numbering.

## TCP Segment Format

- **ترويصة TCP (TCP Header).**

● المعطيات Data.

يبين الشكل التالي الحقوق التي يتألف مقطع بروتوكول التحكم بالنقل:



الشكل (28): مقطع لبروتوكول التحكم بالثقل TCP

- **Source Port (16 bit):** رقم المنفذ المصدر لهذا المقطع.
  - **Destination Port (16 bit):** رقم المنفذ الوجهة (الهدف).
  - **Sequence Number (23 bit):** يستخدم هذا الرقم لضمان تتابع صحيح للبيانات القادمة. وهو الرقم المخصص لأول بايت (byte) من المعطيات في الرزمة المرسلة.
  - **Acknowledgement Number (32 bit):** يدل على رقم الباييت التالي المتوقع استقباله حالياً , حيث أن المستقبل استقبل كل البايتات التي قبل هذا الباييت بنجاح.
  - **Header Length (HLEN) (4 bit):** يحوي على رقم صحيح يدل على طول ترويسة Segment (عدد الكلمات بطول عدد الكلمات بطول 32-bit ضمن الترويسة).
  - **Reserved (6 bit):** محجوز من أجل الدراسات المستقبلية.
  - **Code bits (6 bit):** تمثل تفعيل أو إلغاء تفعيل توابع التحكم أو (الميزات) كأعداد الجلسة أو إنهاؤها على سبيل المثال , يتألف من 6 بتات وكل منها من أجل تفعيل ميزة أو إلغائها.
- هذه الميزات هي:

**URG:** للدلالة على أن المعطيات مستعجلة.

**ACK:** للدلالة على أن المقطع يحتوي على ACK.

**PSH:** يجب على المستقبل إرسال هذه المعطيات أسرع ما يمكن إلى التطبيق.

**RST:** إنهاء الوصلة (link) بشكل غير نظامي.

**FIN:** انتهى المرسل من إرسال المعطيات.

- **Window (16 bit):** يستخدم هذا الحقل للتحكم بالدفق ويحوي على عدد البايتات التي يستطيع المستقبل استيعابها في خزانة التخزين (Buffer) المخصصة للمعطيات المستقبلية ابتداءً من البايت الموجود في الحقل ACK.

- **Checksum (16 bit):** يستخدم للتحقق من صحة وصول الترويسة والمعطيات .

- **Urgent Pointer (16 bit):** تدل على رقم آخر بايت في المعطيات المستعجلة.

- **Options:** للدلالة على الخيارات المطلوبة.

- **Data:** تمثل المعطيات.

ويوجد **TCP(TCP Socket)** ويتم تحقيق ذلك برمجياً عن طريق إنشاء مقبس نوعين من المقابس:

1- مقبس المُخدم **server socket**: تستخدم من أجل نقل المعلومات وتقوم بالتتصت على

طلبات الاتصال القادمة إلى المخدم وتدعى **passive socket**.

2- المقبس العادي **ordinary socket**: عندما يستلم مقبس المخدم ( **server socket**)

( من **ordinary socket** طلب الاتصال من الزبون , يقوم ببناء مقبس عادي )  
أجل تخديم

الزبون , وبعدها يقود مقبس المخدم إلى الاستماع على طلبات الاتصال.

• يجري إنشاء **server socket** عن طريق التعليمة:

**ServerSocket(int port);**

حيث **port** هو رقم بوابة المخدم التي يتتصت عليها

**ServerSocket welcome= new ServerSocket(5000);**

**accept()** يقوم المخدم بقبول طلب الاتصال عن طريق التابع

**Socket client=welcome.accept();**



- يجري إنشاء **ordinary socket** عن طريق التعليمة:

**Socket(InetAddress server,int port);**

**InetAddress server**: عنوان ( ip ) المخدم.

**Port**: رقم بوابة المخدم التي يستمع عليها.

**Socket skt = new Socket("192.168.0.1",5000);**

ويعتبر هذا البروتوكول من بروتوكولات النقل الموثوقة في عملية نقل البيانات.

# الفصل السابع

## الإجراء العملي

## 1.7- شبكة كشف الأحرف :

تم تنفيذ هذه المرحلة وفق الخطوات التالية :

- 1- جمع بيانات صور لتدريب الشبكة (data) .
- 2- معالجة البيانات.
- 3- بناء معمارية الشبكة العصبونية.
- 4- تدريب الشبكة على البيانات المعالجة مسبقاً .
- 5- اختبار دقة الشبكة .

### أولاً: جمع البيانات data collecting :

قد تم تصوير البيانات لتسعة عشر صنف categories :

A, B, C, D, F, G, H, I, K, L, O, P, Q, R, U, W, X, Y, Nothing

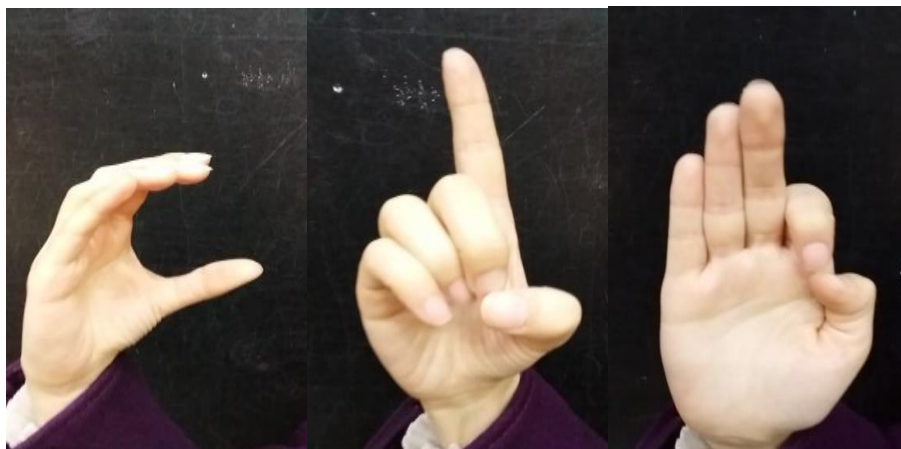
وقد قمت التقاط ما بين 200 إلى 250 صورة لكل صنف وسطياً من خلال كاميرا موبايل دقتها 1836 x 3264 , وتمت مراعاة ظروف الإضاءة والبيئة المحيطة , حيث جرى التصوير في مدرج من مدرجات الجامعة مقابل اللوح الأسود لسهولة تعريب الصورة لاحقاً , وقد بلغ حجم البيانات 4725 صورة .



Nothing

A

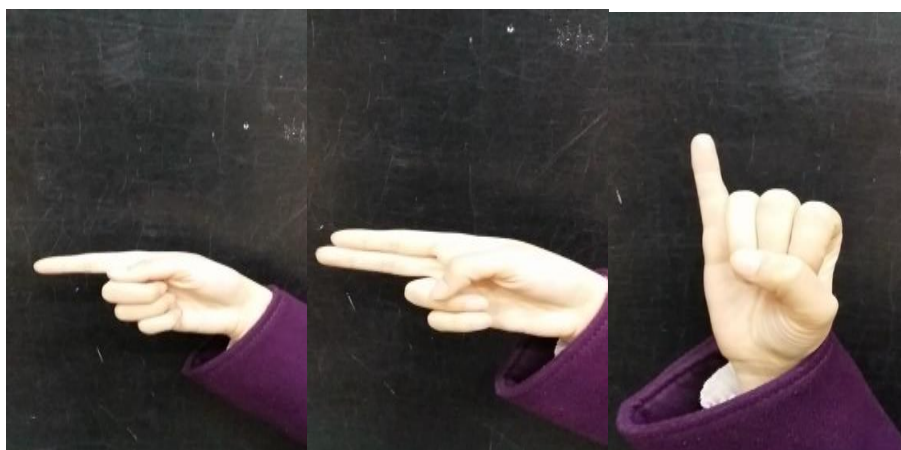
B



C

D

F



G

H

I



K

L

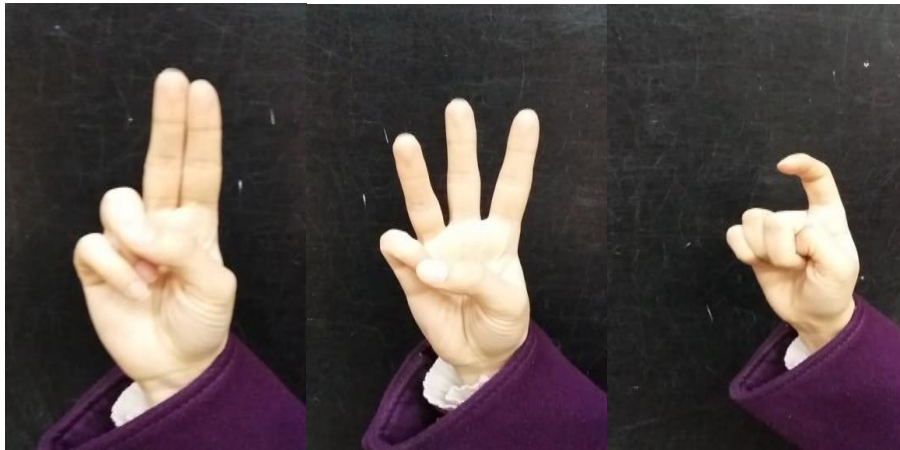
O



P

Q

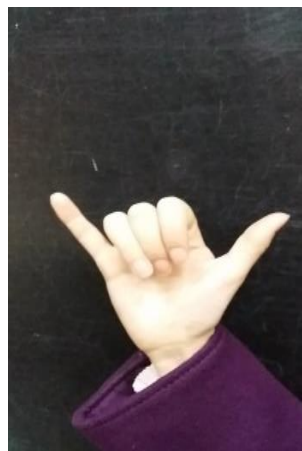
R



U

W

X



Y

## ثانياً معالجة البيانات :

- 1- تحويل الصورة إلى اللون الرمادي .
- 2- تعتيب الصورة .
- 3- ضغط لصورة ليصبح حجمها 64x64 .
- 4- خلط الصور قبل تقديمها للشبكة.

## ثالثاً: بناء معمارية الشبكة العصبونية :

طبقات الشبكة العصبونية :

conv2D (3x3) +ReLU

conv2D (3x3) +ReLU

MaxPolling 2x2

Dropout 0.25

conv2D (3x3) +ReLU

conv2D (3x3) +ReLU

MaxPolling 2x2

Dropout 0.25

conv2D (3x3) +ReLU

conv2D (3x3) +ReLU

MaxPolling 2x2

Dropout 0.25

Flatten

Dense 128

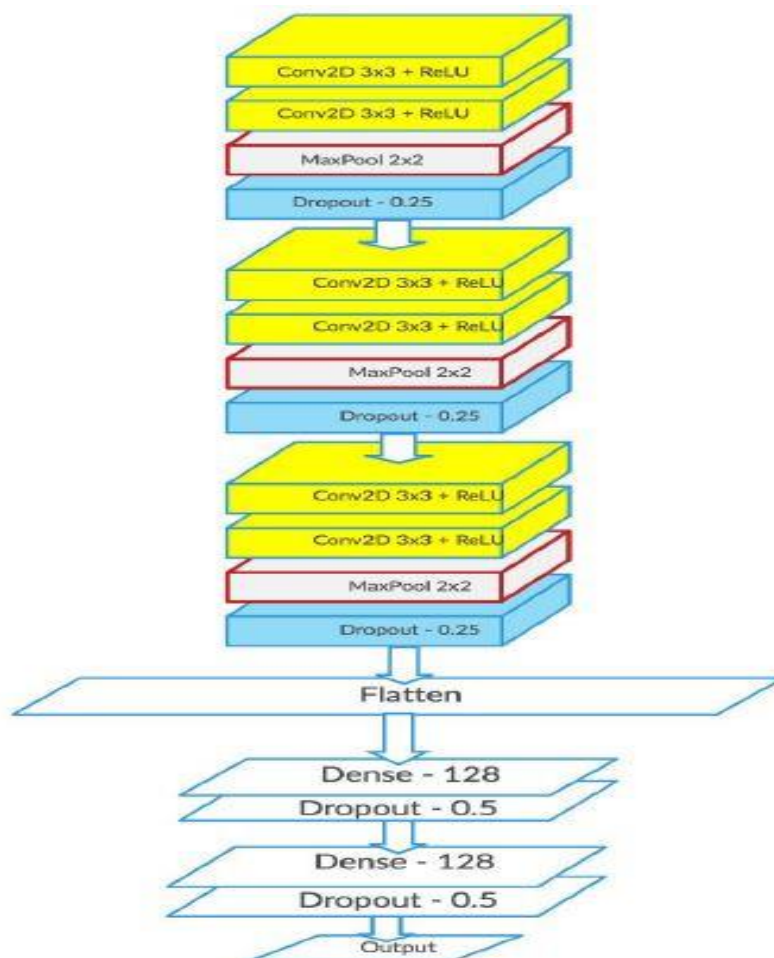
Dropout 0.5

Dense 128

Dropout 0.5

Dense 19

Softmax



الشكل (29): معمارية شبكة كشف الأحرف

## رابعاً : تدريب الشبكة العصبونية :

تم استخدام تابع خطأ `sparse_categorical_crossentropy` , adam optimizer  
batch size=32, epochs=30, validation split=0.35

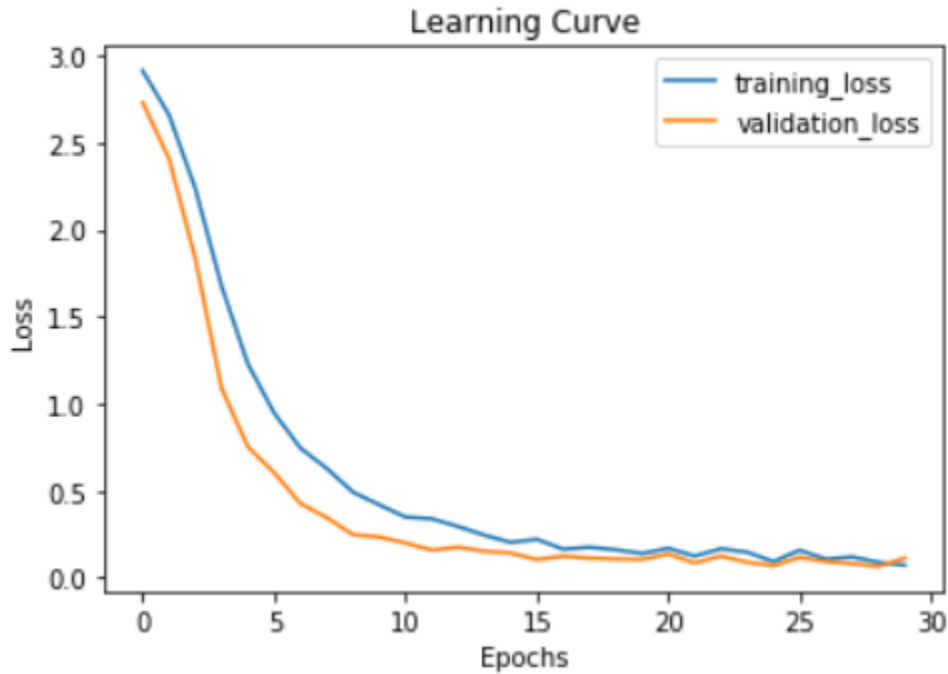
وتم تحقيق النسب التالية :

loss: 0.0976

accuracy: 0.9733

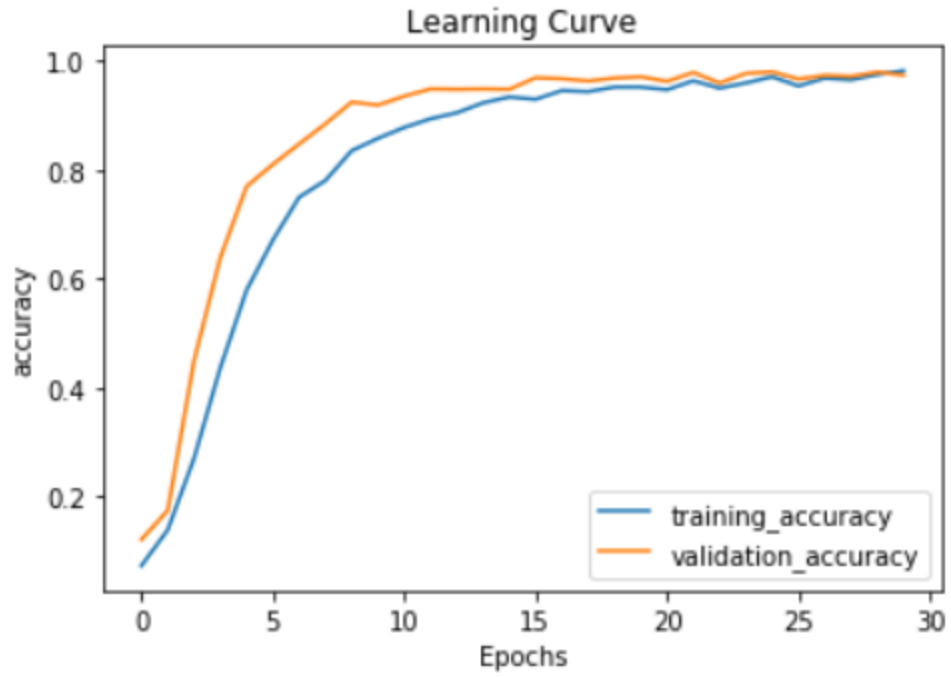
validation loss: 0.1287

validation accuracy: 0.9722



الشكل(30): منحنى تغيرات الخطأ خلال التدريب





الشكل (31): منحنى Accuracy

## 2.7- شبكة التعرف على الإيماءات :

و لإنجاز هذه الشبكة تم اتباع الخطوات التالية :

- 1- جمع بيانات لتدريب الشبكة
- 2- معالجة البيانات
- 3- تصميم الشبكة و اختيار معماريتها
- 4- تدريب الشبكة و اختبار دقتها

### أولاً: جمع البيانات لتدريب الشبكة:

تم تصوير فيديو لكل حركة من حركات لغة الإشارة مدة الفيديو 6 ثواني لكل حركة و ذلك في ظروف إضاءة مناسبة و تم ذلك في احد مدرجات الجامعة حيث تم تصوير 5 إيماءات و لكل حركة 31 فيديو مختلف من أجل تدريب الشبكة على تصنيفها و هي :

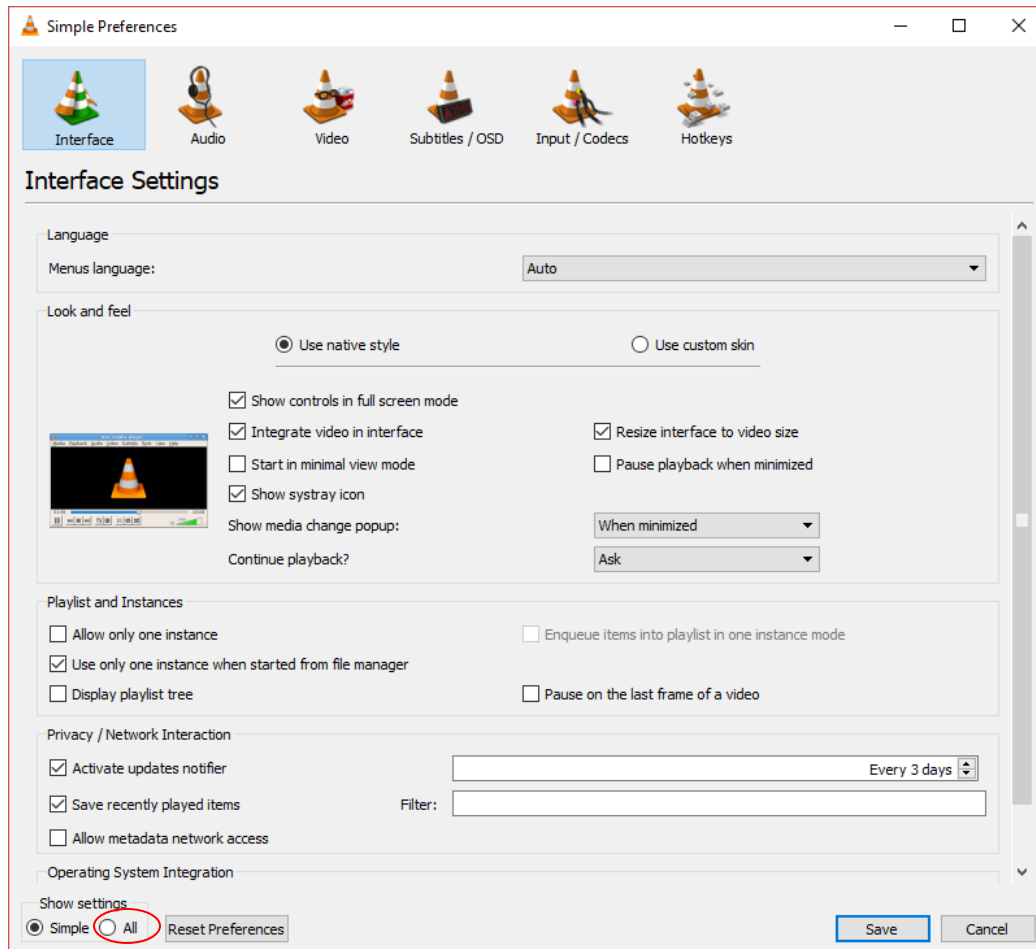
[ no gesture, colic, Radiograph , my , yes ]

### ثانياً: معالجة البيانات :

بدايةً تم تقطيع الفيديوهات إلى إطارات متتالية (frames) بمعدل تقطيع 6frames/sec باستخدام برنامج VLC media player كالتالي :

Tools → preferences

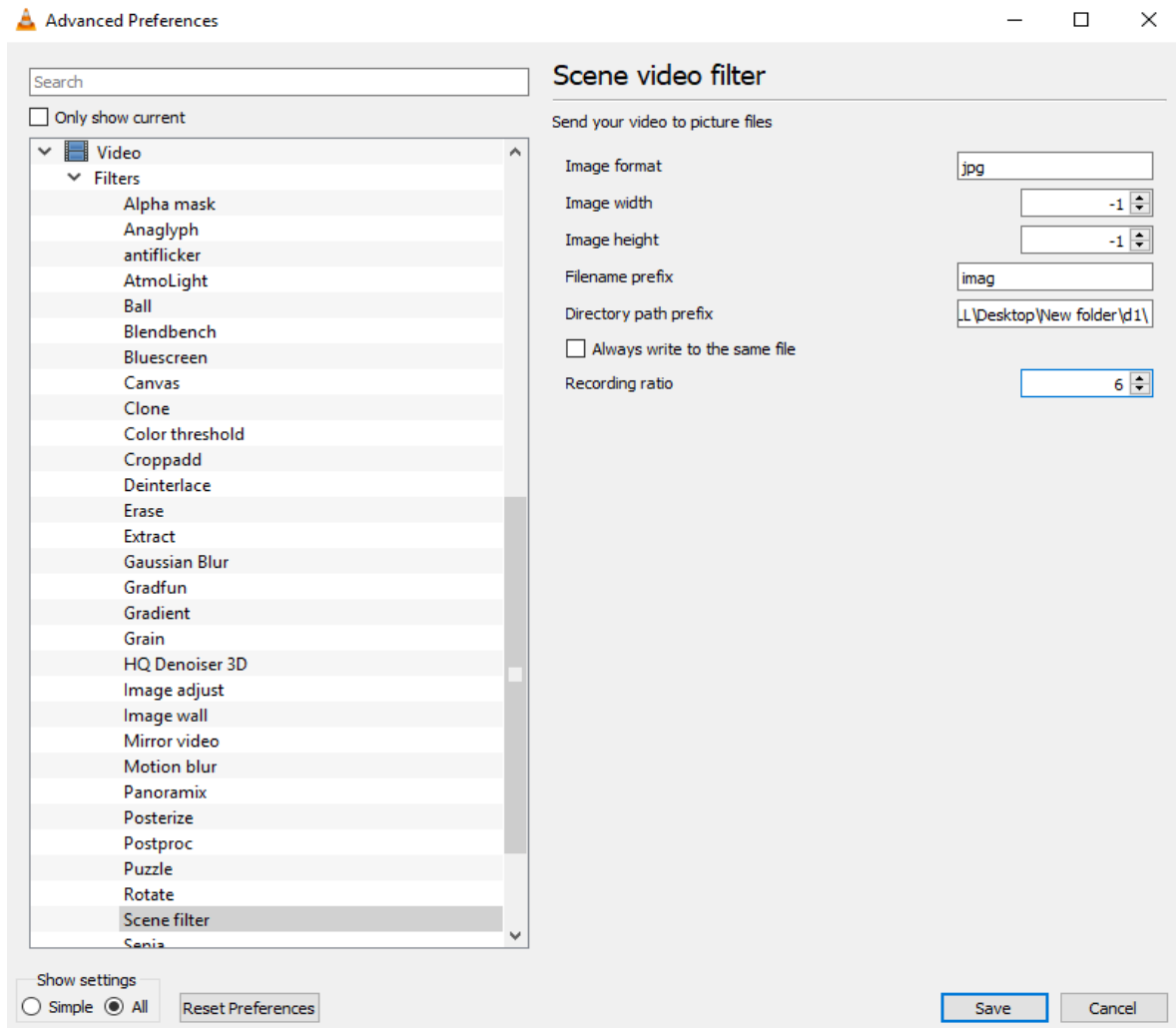
تظهر لنا الواجهة التالية نضغط على خيار all



بعد الضغط على خيار all تظهر الواجهة التالية نختار من video

Video → filters → scene filter

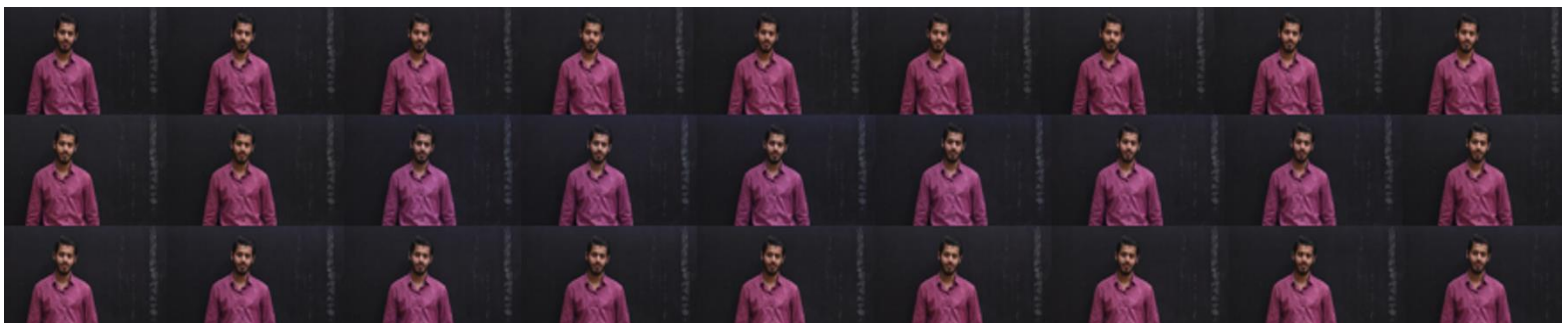
ضمن هذه الواجهة يمكن تحديد معدل التقطيع و مسار الملف الذي سيتم تخزين الإطارات فيه .



بالضغط على save يتم حفظ الاعدادات ..

صور العينات بعد التقطيع :

• No gesture :



: Colic •



: My •



: Radiograph •



• Yes :



الخطوة التالية هي تهيئة هذه الإطارات لدخولها إلى الشبكة العصبونية و ذلك من خلال تحويل النظام اللوني من rgb إلى gray scale و من ثم إعادة تقييس الصور لحجم 64\*64 بالإضافة إلى تنعيم الصورة باستخدام المرشح الغاوسي .

### ثالثاً : تصميم الشبكة و اختيار معماريتها :

اعتماداً على التحليل و التجريب تم اختيار المعمارية التالية :

- Input layer
- 1 convolution3D layer
- MaxPooling3D(2,2,2)
- Dropout(0.25)
- 2 convolution3D layer
- MaxPooling3D(1,2,2)
- Dropout(0.25)
- 2 convolution3D layer
- MaxPooling3D(1,2,2)
- Dropout(0.25)
- 2 convLSTM2D layer

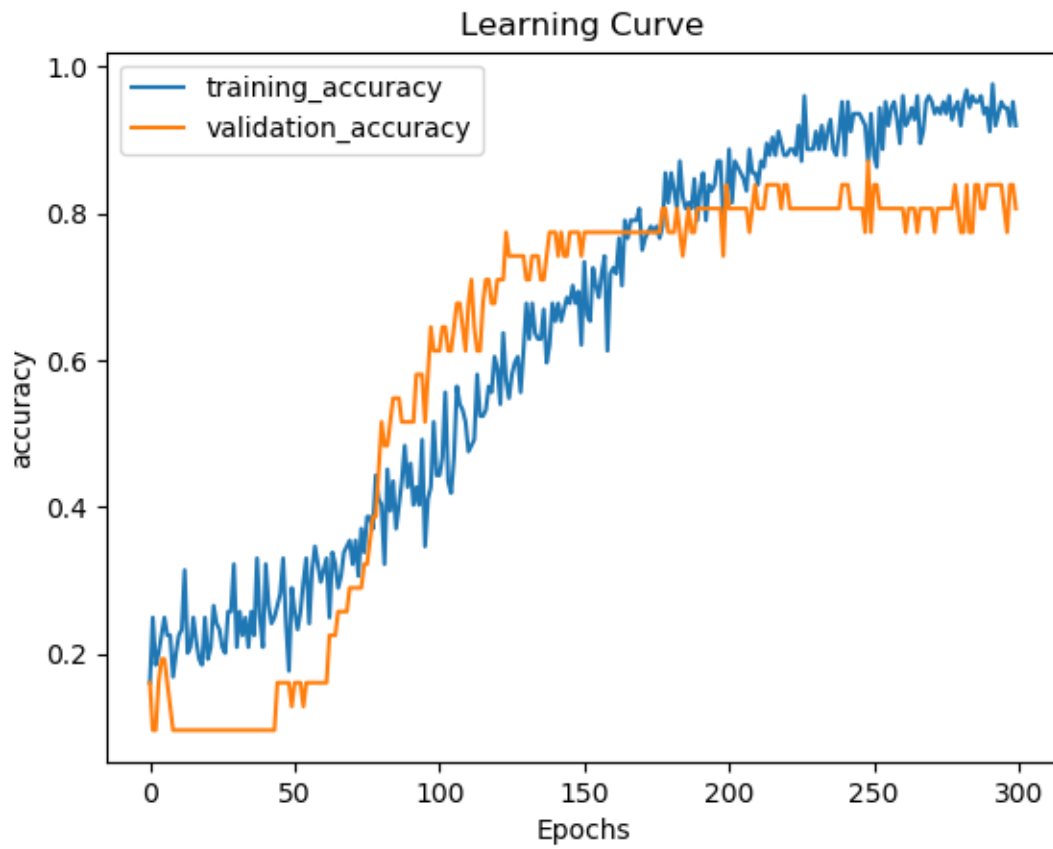
- Flatten
- Dense(128)
- Dropout(0.5)
- Dense(128)
- Dropout(0.5)
- Output layer

### رابعاً : تدريب الشبكة و اختبار دقتها :

في عملية التدريب تم اختيار المتحولات الخاصة بعملية التدريب كالتالي :

- Loss: categorical\_crossentropy
- Optimizer: SGD with learn rate = 0.01
- batch size = 32
- number of epoch = 300
- shuffle
- testing size = 0.2 from all dataset

و تم الحصول على النتائج التالية :



الشكل (33): منحنى Accuracy

### 3.7- قسم الإرسال وتطبيق Android:

- أيقونة التطبيق:





الشكل (34): أيقونة تطبيق Android

تطبيق android يمثل المخدم ويتألف من الواجهات التالية:

• الواجهة الرئيسية:

يمكن للمستخدم من خلالها الانتقال إما إلى:

- الواجهة التي يتم فيها فتح كاميرا الهاتف المحمول.

- الواجهة التي يتم فيها ضبط الإعدادات.

-الواجهة التي تحوي معلومات عن التطبيق.

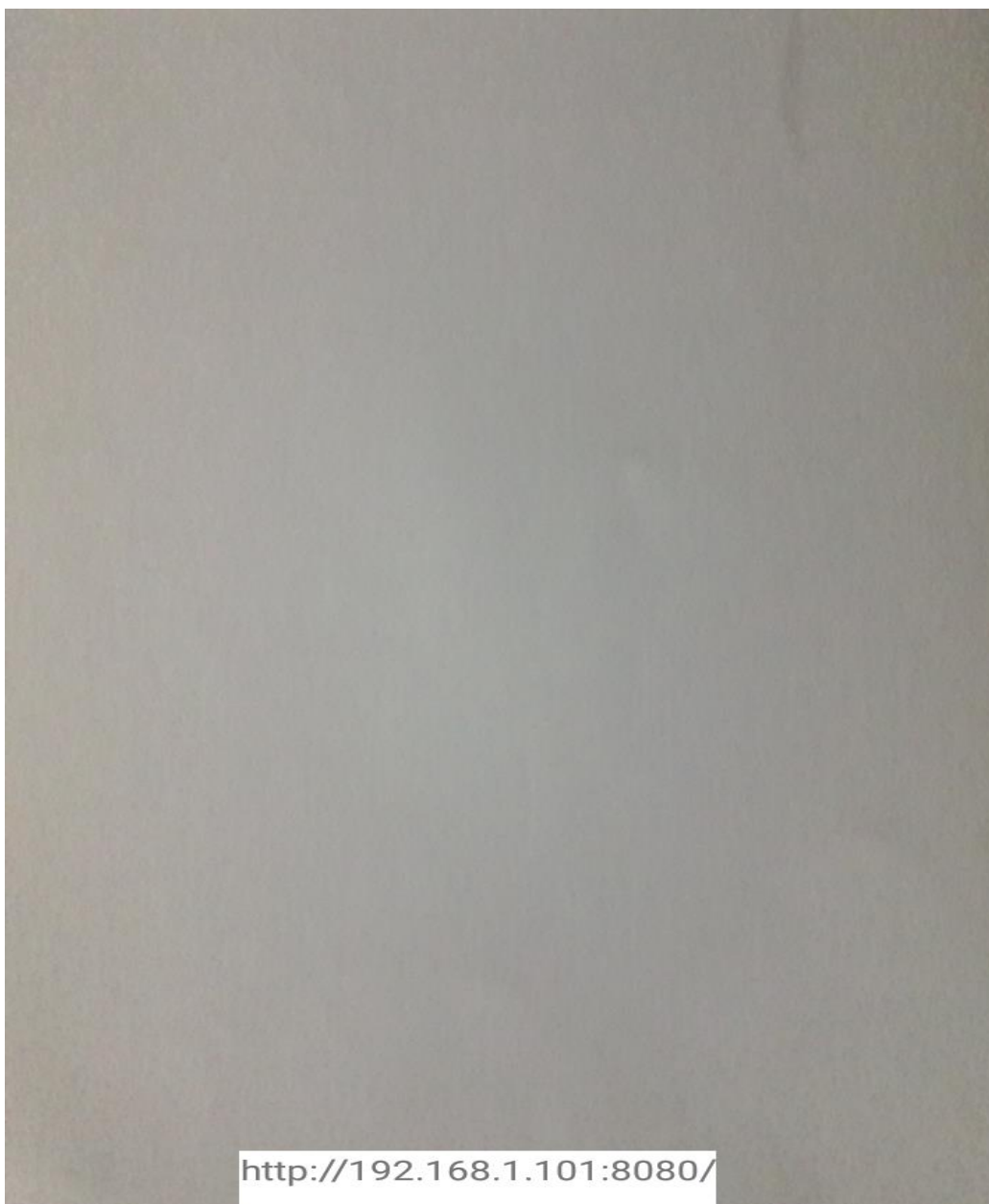


الشكل (35): الواجهة الرئيسية

#### ● واجهة البث:

يتم من خلالها فتح كاميرا الهاتف المحمول و أخذ الإطارات (frames) منها ,

و رفعها على ip الهاتف المحمول الموجود أسفل الواجهة ليتم من خلاله إرسال البث المباشر إلى الحاسب لإجراء المعالجة المناسبة للحركة .

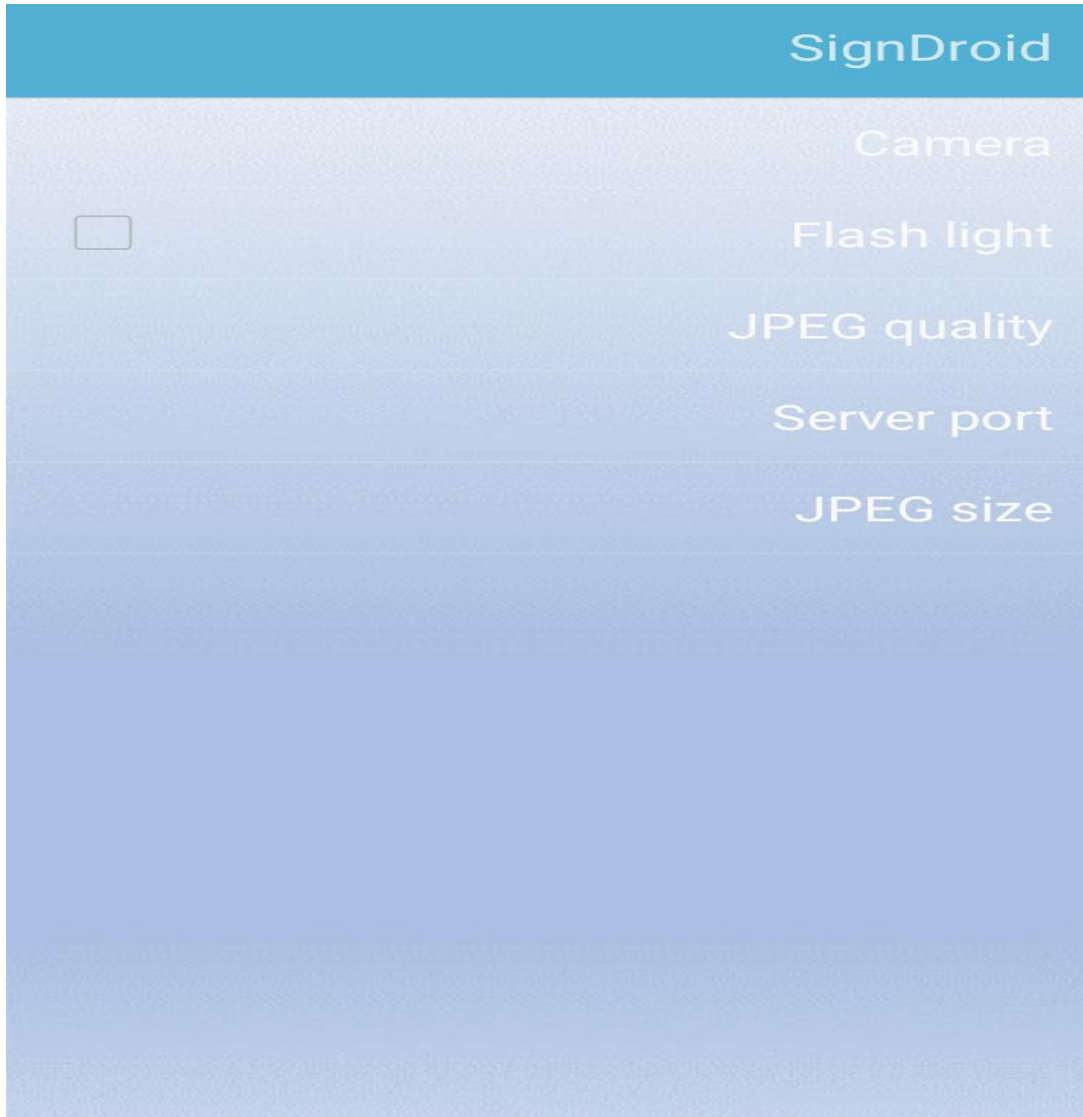


الشكل (36): واجهة البث

## واجهة الإعدادات:

يمكن للمستخدم من خلالها :

- تشغيل الكاشف الضوئي.
- تحديد الكاميرا الأمامية أو الخلفية.
- رقم البوابة المراد اعتماده للبحث.
- حجم الصورة.
- دقة الصورة.



الشكل (37): واجهة الإعدادات

● واجهة تعريفية بالبرنامج:

تجوي هذه الواجهة ملخص عن هدف المشروع وبعض التفاصيل عن استخدامه

ومعلومات عن فريق العمل.

The main goal of application is to help deaf and mute people communicate with others by taking frames from the mobile's camera and sending it to a device that acts as a server to process these images

This project has developed by:

Mhd Hasan Alhallak

Dina Alsalkhade

Suad Yacoub Agha

Farah Shams Aldeen

الشكل (38): واجهة تعريفية بالبرنامج

### الصعوبات :

1. صعوبة تنصيب TensorFlow-GPU فاضطررنا للعمل على TensorFlow-CPU مما أدى إلى زيادة في زمن التدريب .

2. صعوبة في جمع عدد كبير من البيانات لتدريب شبكة الإيماءات.

3. صعوبة في الاختبار على الهاتف المحمول مباشرةً باستخدام TensorFlow Lite فلجاناً إلى طريقة server/client.

4. عدم إمكانية إرسال إطارات الصور (frames) بالزمن الحقيقي باستخدام socket , و كان الحل رفع إطارات الصور بشكل مباشر إلى IP باستخدام HTTP .

## الآفاق المستقبلية :

1. التعرف على عدد أكبر من الإيماءات .
2. الاختبار على الهاتف المحمول باستخدام TensorFlow Lite.

3. إدخال NLP(Natural Language Programming) على شبكة  
الإيماءات .