



الجامعة الافتراضية السورية

SYRIAN VIRTUAL UNIVERSITY

(التعلم العميق للكشف الآلي عن تعابير الوجه والمشاعر)

إشراف: د. أميمة دكاك.

مشرف مساعد: م. منار منعم.

إعداد:

تاج الدين الموصلی

محمد تميم علیوی

F20_PR2_C4

الملخص **Abstract**

يقدم المشروع بناء نماذج تعلم عميق لتصنيف صورة معطاة لوجه بشري إلى واحدة من المشاعر الأساسية السبعة عند الإنسان (الغضب - الاشمئاز - الخوف - السعادة - الحزن - الحياديّة - المفاجأة).

وقد تم عرض المشروع ضمن مقدمة ومجموعة فصول، حيث تم في المقدمة استعراض بسيط للمشاعر الأساسية عند الإنسان وما هيها وصولاً إلى فكرة المشروع التي تتضمن بناء نماذج للتعلم العميق، أما الفصول فقد تم عرض مراحل المشروع فيها كما يلي:

الفصل الأول: تعريف المشروع من حيث الهدف والشروط والمتطلبات التي احتاجها.

الفصل الثاني: الدراسة المرجعية والتي تشمل دراسة عن مختلف الطرق لتنفيذ نموذج تعلم عميق للكشف عن المشاعر، بالإضافة إلى شركات قائمة على فكرة مشروعنا.

الفصل الثالث: الدراسة التحليلية للمشروع، حيث تضمنت عناوين (التخطيط وتحليل المتطلبات) حيث تم تحت عنوان التخطيط استعراض الأسباب الموجبة للمشروع وتحليل الجدوى إضافة إلى التخطيط لإدارة المشروع من حيث مراحله وموارده مروراً بإدارة الموارد وتم عرض خطة المشروع باستخدام مخطط غانت "MS Project" انتهاءً بمصفوفة المسؤوليات.

الفصل الرابع: التعريف بالتقانات التي قمنا باستخدامها.

الفصل الخامس: تقرير عن الطرق التي تم اتباعها في محاولة حل هذه المسألة.

الفصل السادس: شرح الاستخدام.

الفصل السابع: الخاتمة التي تتضمن العمل ككل وتم استعراض الآفاق المستقبلية للتطوير المستقبلية للمشروع.

الفهرس

5	مقدمة:
7	الفصل الأول
8	الهدف من المشروع:
8	الشروط والمتطلبات:
9	الفصل الثاني
10	الفصل الأول
11	مقدمة:
12	1. استخراج ميزات المشاعر متعددة الأشكال
14	DSAN.2
22	3. AdaBoost-KNN) مع التحسين المباشر للتعرف على المشاعر الديناميكية
35	الفصل الثاني
39	الفصل الثالث
40	أولويات المشروع:
40	ادارة المشروع:
40	منهجية العمل:
41	اسباب استخدام نموذج الشلال:
41	ميزات نموذج الشلال:
42	هيكل تجزئة المشروع:
43	أولاً: مرحلة التخطيط:
43	ثانياً: مرحلة التصميم:
43	ثالثاً مرحلة الاختبار:

44	رابعاً: مرحلة التوثيق:
44	الدراسة التخطيطية:
45	إدارة المخاطر:
47	مخطط الأنشطة الإجرائية:
49	مصفوفة المسؤوليات:(RM) <i>Responsibility Matrix</i>
51	الفصل الرابع
66	الفصل الخامس
68	الملف الأول
83	الملف الثاني
91	الملف الثالث
96	الملف الرابع
106	الملف الخامس
109	الفصل السادس
117	الفصل السابع
117	الخاتمة
118	الخاتمة:
120	المراجع:

مقدمة:

يقدم تعريف المشاعر [1] على أنه التغيرات في الحالات النفسية التي تشمل الأفكار، والسلوكيات التعبيرية للتصرف. إن التركيبة الدقيقة للتغيرات النفسية يتقلب من عاطفة لأخرى، وليس من الضروري أن يكون مسحوب بتصرفات مرافقته له. إن هذه الحالات والسلوكيات المعقدة يتم تشغيلها من قبل حدث يتم تجربته أو إعادة حدوثه¹.

فمثلاً إذا رأيت شخص تحبه، اعتماداً على تصرفه من الممكن أن تبقى ذو تعابير حيادية طبيعية، أو سوف تصاب بالخجل ويصبح لون وجهك أحمرأ.

وبالاعتماد على مثل هذا المثال الذي سبق، يمكن إعطاء نوع مشابه من المشاعر ألا وهي: الغضب، والاشمئزاز، والخوف، والسعادة، والحيادية، والحزن، والمفاجأة، والعديد غيرها. وهناك أيضاً مشاعر هي أقل وضوحاً، وهي لا تظهر تغيرات في التعابير العاطفية ولا يستطيع الإنسان أن يعبر عنها دائماً، مثل الندم بعد اتخاذ قرار أو اتخاذ قرار انتضح أنه سيئ، هنا من الممكن أن يشعر الإنسان بالندم الشديد، لكن هذه التجربة الذاتية عادةً لا تكون مصحوبة بأي تغيرات في تعابير الوجه أو السلوكيات².

التعابير والمشاعر:

يمكن تحليل المشاعر من تعابير الوجه ومن أجزاء أخرى من الجسم[2]. إذا كان أحدهم يشعر بغضب شديد، والحزن والخوف والمفاجأة، فإننا نلاحظ التغيرات في تعابيراته على صورة وجه معين ونحاول أن نفرق فيما بين هذه المشاعر. التقلب للعديد من صور الوجه في المشاعر يساعدنا في فهم ما يشعر به أحدهم بدون التواصل غير اللفظي. وفي بعض الأحيان، يتم التجاذل في التواصل غير اللفظي للعاطفة هو أكثر أهمية وفعالية من الاتصال اللفظي. هذا ينطبق على التواصل بين الأشخاص حيث لا يستطيع أحدهم أو كلاهما التواصل لفظياً لأنهم ما قبل اللغة (prelingual) كما هو الحال في حالة الرضع أو الصم أو ببساطة غير قادرين على التحدث مع بعضهم البعض³.

[3] بين أن هناك علاقة وثيقة بين المشاعر وسلوك الوجه، لقد قاموا باقتراح نظام له القدرة على التعرف على تعابيرات الوجه. لقد أظهروا طريقتين للبحث. تعتمد الطريقة الأولى على فكرة داروين التي تعتبر أن المشاعر التي يتم التعبير عنها هي عالمية ومستقلة أيضاً عن الثقافة. Ekman وأخرون التقىوا صوراً للوجه تم تحديدها على أنها تعبير بشكل صريح عن بعض المشاعر، وأيضاً قاموا بتقييمها لأشخاص في مجموعة متنوعة من الثقافات الأخرى⁴.

عن المشاعر:

من أكثر الأبحاث التي برزت في موضوع تمييز المشاعر من تعابير الوجه جاءت من عالم النفس (Paul Ekman)

¹ Human emotion recognition from face images, CHAPTER 1, P.1

² Human emotion recognition from face images, CHAPTER 1, P.1

³ Human emotion recognition from face images, CHAPTER 1, P.2

⁴ Human emotion recognition from face images, CHAPTER 1, P.2

[3] ، الذي كان رائداً في مجال الأبحاث العلمية المتعلقة بالتعرف على المشاعر في ستينيات القرن الماضي. قدم فريقه من الباحثين للأشخاص الخاضعين للاختبار صوراً لوجوه تظهر تعبيرات مشاعر مختلفة. الخاضعون للاختبار عرفوا المشاعر العاطفية في كل صورة، بناءً على قائمة محددة مسبقاً من المشاعر المحتملة. في خلال هذه الدراسات، وجد Ekman أنه بين كل الأعضاء الخاضعين للاختبار سواءً كانوا من خلفيات ثقافية شرقية أو غربية اتفاقاً كبيراً عندما يتعلق الأمر باختيار المشاعر التي تتوافق مع تعبير الوجه. لقد عرّفوا المشاعر العالمية على أنها تلك التي تشير إلى السعادة، والاشمئزاز، والغضب، والحزن، والمفاجأة، والخوف، والحيادية. من خلال العمل مع Friesen وضع Ekman أساس نتائج دراسته المرتبطة بقبائل فور في بابوا غينيا الجديدة (Fore tribesmen in Papua New Guinea) التي لم يستطع أعضاؤها تعلم معنى المشاعر من التعرض لوسائل الإعلام. وعوضاً عن ذلك، قاموا بالتصريف والتعبير عن مشاعرهم بالغرائز التي قد ورثوها بدون أن يتم تعليمهم ماهية كل نوع من المشاعر والتعبير الخاص به، مما دفع كل من Ekman & Friesen [3] إلى الوصول إلى نفس فكرة داروين، إلا وهي عالمية المشاعر⁵.

ومع تقدم التكنولوجيا والعلوم، تبقى مسألة التعرف على المشاعر من صور الوجوه من المسائل التي يتم تطويرها باستخدام الكثير من التقنيات الحديثة مثل تقنيات التعلم العميق والشبكات العصبية، وتتنوع التطبيقات والطرق التي يتم فيها تنفيذ مثل هذه المسائل، وتکاد أن تكون لا تحصى.

في مشروعنا هذا، سوف نستعرض قدر ما نستطيع من خوارزميات وطرق لتنفيذ هذه المسألة بالإضافة إلى تفاصيل ونتائج كل منها.

الفصل الأول

تعريف المشروع

الهدف من المشروع:

يهدف المشروع إلى تقديم نماذج للتعلم العميق لمحاولة حل مسألة التعرف على المشاعر من خلال الوجوه، ومحاولة الحصول على أعلى دقة للتعرف على المشاعر في صور الوجوه، والمشاعر المطلوبة هي: الغضب - الاشمئزاز - الخوف - السعادة - الحزن الحياديّة - المفاجأة.

الشروط والمتطلبات:

يشترط بالمشروع تأمين ما يلي:

- استعراض العديد من الطرق لحل هذه المسألة وتقديم العديد من الطرق السابقة لتنفيذ هذا النوع من النماذج.
- استعراض العديد من الطرق لحل هذه المسألة.
- تقرير عن الطرق التي تم اتباعها في محاولة حل هذه المسألة
- بالإضافة إلى:
 - تجربة المستخدم للنظام بالزمن الحقيقي.
 - توفير كافة المعلومات المتعلقة بتنفيذ هذا المشروع.

الفصل الثاني

الدراسة المرجعية

وتقسم إلى:

.1 حلقة بحث باللغة العربية عن العديد من طرق تنفيذ وخوارزميات متعلقة بهذا المشروع.

.2 شركات تعتمد على فكرة مشروعنا.

القسم الأول

الدراسة المرجعية

حلقة بحث باللغة العربية عن العديد من طرق تنفيذ وخوارزميات متعلقة بهذا المشروع.

مقدمة:

إن تكنولوجيا التفاعل بين الإنسان والروبوت قد تحولت تدريجياً من التركيز على الحاسوب إلى التركيز على الإنسان، حيث أصبح التفاعل الطبيعي بين الإنسان والروبوت اتجاهها جديداً يدخل في تطوير تكنولوجيا التفاعل بين الإنسان والروبوت. ومن أهم هذه التفاعلات، هي تفاعل المشاعر، وهي جوهر وأساس تفاعل الإنسان الطبيعي والروبوت، والتي تجعل الروبوتات قادرة على التواصل العاطفي، بانسانية أكثر من خلال معالجة معلومات المشاعر [4] في عملية التفاعل بين الإنسان والروبوت. وبظهور البيانات الضخمة (BIG DATA)، فإن ابتكار الخوارزميات التي تحسن قدرة الحاسوب بشكل عام والحوسبة بشكل خاص، وتطور شبكة من المنشآت التي تقود تطوير الذكاء الاصطناعي. الذكاء قد أصبح اتجاهها مهم من أجل تطوير التكنولوجيا والصناعة [5]. والآن، فإن ثورة الروبوت قد دخلت عصر كل من الشبكة العنكبوتية، والذكاء الاصطناعي، والمشاعر، والتي تتطلب أن يكون لدى الروبوت القدرة على الإدراك العاطفي للمشاعر [6].

الروبوتات العاطفية الذكية تظهر بشكل متزايد في حياتنا اليومية. بناءً أنظمتهم عادةً مبني على روبوت عادي له صفات إنسان، وإطار استرجاع معلومات للمشاعر، إطار فهم للمشاعر، إطار تفاعل للمشاعر خاص بالروبوت، وأخيراً شكل مشابه للإنسان، نظام روبوت بمشاعر متداخلة مع مشاعر البشر، وبقدرة معرفية [7]. على سبيل المثال: يمكن للروبوت (Buddy) وهو أول روبوت ذو مشاعر مخصص لمصاحبة العائلات وتم إطلاقه من قبل شركة (Blue Frog Robotics) يستطيع بطبعية التعبير عن مشاعره وأن يتبعها بالتفاعل المناسب مع أفراد الأسرة [8]. الروبوت الذي يشبه الإنسان صوفيا (Sophia)، تم تطويرها من قبل (Hanson Robotics)، وكانت أول روبوت في التاريخ تحمل جنسية بلد (البلد هو السعودية)، يمكن لهذا الروبوت عرض أكثر من 62 تعبيراً وفهم اللغة المحكية وتذكر التفاعلات مع البشر [9]. ويمكن (Erica)، وهو روبوت ذكي تم تطويره من قبل فريق البروفيسور Hiroshi Ishiguro) في جامعة أوساكا في اليابان، وأيضاً من قبل فريق بحثي من جامعة كيوتو (Osaks & Kyoto)، يمكن لهذا الروبوت التحدث بطلاقة مع الناس بأصوات وتعبيرات قريبة جداً للبشر. بالإضافة لذلك، فإن (Erica) في مرة من المرات عملت كمذيعة وقدمت نشرة أخبار في محطة تلفزيونية يابانية في عام 2018 [10].⁷

إن فهم المشاعر يعتمد على تحليل ارتباط النوايا العاطفية، ويعتمد أيضاً على دراسات أخرى حول كيفية اكتشاف المزيد من النظريات التفصيلية، والأساليب والتقنيات المتعلقة بفهم النوايا من المعلومات السطحية للمشاعر العميق، الهدف منها هو فهم نوايا المستخدم الشخصية بناءً على مشاعرهم، ومعلومات الاتصال السطحية، وسيناريوهات محددة، وفي النهاية تحقيق انسجام طبيعي من أجل فهم الآلة لطبيعة مشاعر الإنسان [11]. وفي حلقة البحث هذه سنتناول العديد من الخوارزميات والطرق والحالات التي تجعل الروبوت أو الحاسوب يفهم مشاعر الإنسان.

1. Multi-modal Emotion Feature Extraction.

1. استخراج ميزات المشاعر متعددة الأشكال.

يعد استخراج ميزات المشاعر متعددة الأشكال قسم لا غنى عنه في التعرف على المشاعر. من أجل الاستخدام الفعال لمعلومات المشاعر في التعرف على المشاعر متعددة الأشكال، يجب اعتماد طريقة استخراج الميزات وفقاً لأنماط المشاعر المختلفة. في هذا القسم من حلقة البحث، تم تقديم طريقة لاستخراج السمات القائمة على التفاصيل التي نريدها في الوجه للتعبير عن المشاعر.

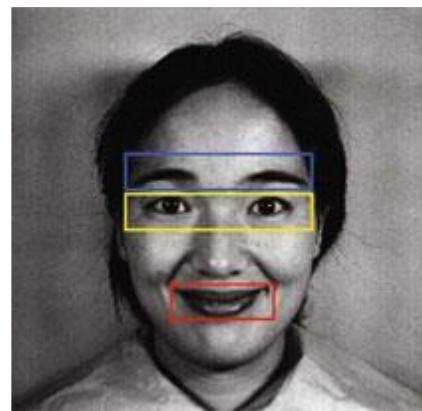
استخراج ميزات تعبيرات الوجه:

قبل استخراج الميزات، نحتاج إلى إكمال بعض العمليات المطلوبة سابقاً، مثل تجزئة الصورة وتحديد أماكن تعبيرات الوجه (*ROI: automatic Region of Interest*)، تعديل حجم الصورة، وتوازن اللون الرمادي، وما إلى ذلك.⁸

في صور الوجه، نسيج وشكل الميزة يتغير في ثلاثة أجزاء رئيسية، وهي: الحاجبين، والعينين، والفم. هذه الأجزاء من الممكن أن تعكس التغيرات الحاصلة في تعبيرات الوجه، ونتيجة لذلك، نستطيع استخدام هذه المناطق في صور الوجه من أجل استخراج الميزات.⁹

بالنسبة لصور الوجه، تم اعتماد قاعدة البيانات (*JAFFE*) الأولى، وتم الحصول يدوياً على إحداثيات الزوايا الأربع للمناطق التي نريدها، وتم قسم الحاجبين، العينين، الفم من تعبيرات صور الوجه هذه. في الجدول (1)، توجد المعلومات عن إحداثيات أربعة زوايا في ثلاثة مناطق في الصورة ومنطقة القطع المستطيلة، كل عمود في مصفوفة الأربع زوايا هذه، هو عبارة عن مصفوفة تمثل أربع نقاط (إحداثيات Y & X) في اتجاه عقارب الساعة، مصفوفة القطع المستطيلة تمثل عرض وارتفاع منطقة قطع كل مستطيل.

من خلال قص هذه المناطق، لا يمكننا فقط تقليل التداخل في معلومات الوجه الناتجة عن تداخل الصورة في الأجزاء غير الحرجة، ولكن أيضاً يقلل كمية البيانات ويحسن سرعة الحوسبة. يتم عرض صورة هذه المناطق المقصوصة في الشكل رقم (1)¹⁰



الشكل رقم (1): مناطق تحديد الميزات في الوجه¹¹

Key parts	Corners coordinates	Clipping region
	$\begin{bmatrix} X_{leftup} \ X_{rightup} \ X_{leftdown} \ X_{rightdown} \\ Y_{leftup} \ Y_{rightup} \ Y_{leftdown} \ Y_{rightdown} \end{bmatrix}$	$[Length, Width]$
Eyebrows	$\begin{bmatrix} 74.21 \ 182.26 \ 182.26 \ 74.21 \\ 100.14 \ 100.24 \ 120.05 \ 120.05 \end{bmatrix}$	$\begin{bmatrix} 108.05 \ 19.91 \end{bmatrix}$
Eyes	$\begin{bmatrix} 74.22 \ 182.25 \ 182.25 \ 74.22 \\ 120.85 \ 120.85 \ 140.12 \ 140.12 \end{bmatrix}$	$\begin{bmatrix} 108.03 \ 19.27 \end{bmatrix}$
Mouth	$\begin{bmatrix} 95.55 \ 162.01 \ 162.01 \ 95.55 \\ 180.03 \ 180.03 \ 210.05 \ 210.05 \end{bmatrix}$	$\begin{bmatrix} 66.46 \ 30.02 \end{bmatrix}$

Comparison of experimental results of emotion recognition

الجدول رقم(1): مقارنة النتائج التجريبية للتعرف على المشاعر.¹²

2. Deep Sparse Autoencoder Network for Facial Emotion Recognition DSAN

تم استخدام شبكة عصبية عميقه (*DNN*) كنموذج تعليمي لتعليم البنية الهرمية لدماغ الإنسان. ومع ذلك، تعاني ال(*DNN*) من مشاكل في كفاءة التعلم والتعقيد الحسابي. لمعالجة هذه المشاكل، يتم استخدام (*DSAN*) لتعلم ميزات الوجه، والتي تأخذ بعين الاعتبار تباين الوحدات المخفية لتعليم الهياكل عالية المستوى. وفي الوقت نفسه، يتم استخدام (*Softmax Regression*) اختصاره (*SR*) من أجل تصنیف میزات التعبیر، فی هذا القسم من حلقة البحث تم اقتراح:

Softmax Regression – Based Deep Sparse Autoencoder Network (SRDSAN)

لتعرف على مشاعر الوجه في التفاعل بين الإنسان والروبوت. وتهدف إلى التعامل مع بيانات كبيرة في مخرجات التعلم العميق من خلال استخدام (*SR*) وغير ذلك، للتغلب على مشاكل **local extrema and gradient diffusion** في عملية التدريب، يتم استخدام عملية ال(*Fine – tunning*) من أجل ضبط أوزان الشبكة الكلية للوصول إلى المستوى الأمثل العالمي، مما يجعل عمق الشبكة العصبية بالكامل أكثر قوة، وبالتالي يعزز أداء التعرف على المشاعر في الوجه¹³.

2.1 - مقدمة:

مع التطور السريع للنظريات والتكنولوجيا في مجال الروبوتات، فإنه من المرغوب به عموماً أن يكون للروبوتات إمكانية التعرف وفهم مشاعر البشر. وفي الوقت نفسه، نظام الخدمة الذكية مع القدرة على التعرف على المشاعر أصبح من أكثر المواضيع جاذبيةً في التفاعل بين الإنسان والروبوت. تعابير الوجه تلعب دوراً مهم في التعرف على مشاعر البشر^[12]، ونتيجة لذلك، التعرف على مشاعر الوجه هو الموضوع الرئيسي في مجال الحسابات المؤثرة.¹⁴

أثناء التعرف على الوجه، تواجه الخوارزميات التقليدية لاستخراج ميزات الوجه قيوداً رئيسية، مثل تنوع وضعيات الوجه وقابلية التغيير، والاختلافات الفردية في بنية الوجه، ودرجة لون البشرة، ويفرض أداء الحاسوب قيوداً على سرعة التدريب، والتأثير على البيئة الخارجية، فمثلاً، الضوء والغرفة وما إلى ذلك، مثل طريقة التدفق البصري^[13]، وطريقة النموذج^[14,15]، تتبع طريقة ميزة النقطة^[16,17]. لذلك، بدأت تقنيات الشبكات العصبية الاصطناعية^[18] باكتساح العالم بشكل كبير في مجال التفاعل بين الإنسان والروبوت عندما دخلت الطريقة التقليدية لاستخراج الميزات والتعرف عليها في عنق الزجاجة.¹⁵

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 3, P.25 ¹³

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 3, P.25 ¹⁴

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 3, P.25+26 ¹⁵

2.2-Softmax Regression-Based Deep Sparse Autoencoder Network:

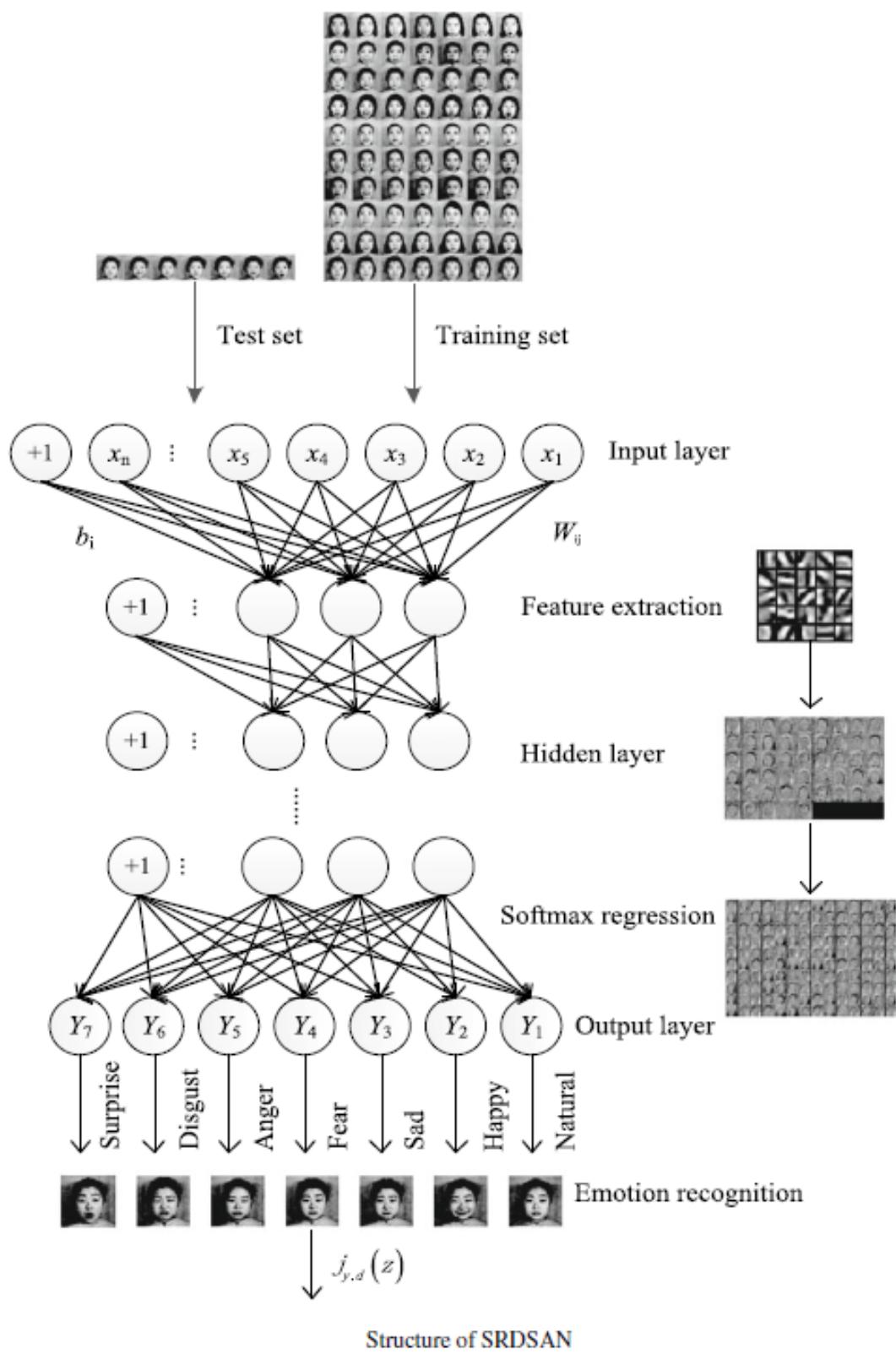
في الشكل رقم (2)، تظهر بنية ال (*SRDSAN*)، والتي تستخدم من *sparse autoencoder network* أجل التعلم العميق، وال (*SR*) من أجل تصنیف میزات التعبیر عن المشاعر. إن خوارزمية (*SRDSAN*)، تكون كالتالي¹⁶:

1. قص الأماكن التي لنا اهتمام بها (مناطق سمات الوجه)
 2. وضع الطبقات (*layers*) طبقة تلو طبقة في الشبكة قبل تدريبها باستخدام ال (*Greedy approach*) من أجل الحصول على مصفوفة أوزان أولية.
 3. الحصول على خرج الشبكة.
 4. تدريب (*SR*) من أجل الحصول على متغيرات النموذج (*parameters*).
 5. تصغیر دالة الكلفة قدر الإمكان (*minimize the cost function as possible*) (*Fine – Tuning*) (*SRDSAN*) بالكامل.
 6. ضبط أوزان ال (*SRDSAN*) بالكامل.
 7. الحصول على تعبير الوجه.
- خوارزمية (*SRDSAN*)¹⁷.
- إن هذه الخوارزمية توضح طريقة ال (*SRDSAN*). أولاً يتم تحديد الحاجبين، والعينين، والفم، على أنهم المناطق التي نريدها ويتم استخراجهم ك سمات لصور لتعبير الوجه. بعدها، يتم إنتاج الأوزان الأولية للشبكة عن طريق ال (*Greedy approach*) ما قبل تدريبها طبقة تلو الأخرى. من أجل تحديد نموذج الشبكة الأفضل، عقد الطبقات المخفية وعدد الطبقات المخفية يتم الحصول عليها عبر تسریع ضئيل متغير. بالإضافة إلى ذلك، فإن ال (*SR*) تستخدم من أجل تصنیف میزات تعبیرات الوجه، ومتغيرات النموذج المثلی لل (*SR*) يتم تدريبها بواسطه طريقة ال (*GD*). وأخيراً كل أوزان ال (*DSAN*) يتم ضبطها من خلال خوارزمية (*BP*)، لجل كل شبكة التعلم العميق أكثر قوة وتعزيز أداء التعرف على مشاعر الوجه¹⁸.

¹⁶ Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 3, P.27

¹⁷ Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 3 Algorithm 3.1, P.27

¹⁸ Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 3, P.27



الشكل رقم (2): بنية الـ (SRDSAN) .¹⁹

2.3-Softmax Regression:

نقدم هنا المصنف (*Softmax*) والذي هو التوسيع للمصنف المنطقي. المصنف المنطقي هو أكثر ملائمة لبعض مشاكل التصنيف الغير خطية، وهو مناسب فقط لمشكلة من نوع تصنيف ثانوي. نتائج التصنيف هي فئات الاحتمال كخرج، يتم تحديد الفئة النهائية من خلال العتبة. إنّ مصنف (*Softmax*) يمكن أن يوسع التصنيف المنطقي، ويجعل له العديد من الصفوف غير الخطية التي تملك القدرة على التصنيف الغير خطى²⁰.

يقوم المصنف الثاني بحساب قيمة احتمالية ثم يقارنها بالعتبة²¹، والتي يمكن أن يتم تحويلها لمسألة تصنيف ثانوي بسيطة. التعبير عن التوابع المنطقية يظهر في المعادلة، حيث $(h_{\theta}(x))$ هي احتمال ال (1)، و θ هي متغير للنموذج²².

$$h_{\theta}(x) = g(\theta^T x) \frac{1}{1 + e^{-\theta^T x}} = p(y = 1 | x; \theta)$$

22

متغيرات نموذج التدريب هي من خلال تعديل وتصغيرتابع الضياع (*loss function*) من خلال التدريب. إن تابع الضياع يعبر عنه في كالتالي²³:

$$J(\theta) = -\frac{1}{m} \left[\sum_{i=1}^m \sum_{j=1}^k 1\{y^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}}} \right]$$

24

التصنيف المنطقي يمكن أن يمتد لكي يتعامل مع العديد من المصنفات، هذا هو مصنف ال (*Softmax*) المتعدد. يتم عرض وضع التوسيع للمصنف المنطقي لاحقاً، حيث أن قيمة الخرج تكون (*k dimensional vector*)²⁵.

$$h_{\theta}(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1 | x^{(i)}; \theta) \\ p(y^{(i)} = 2 | x^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k | x^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}$$

26

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, **CHAPTER 3**, P.31+32²⁰

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, **CHAPTER 3**, P.32²¹

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, **CHAPTER 3 EQ:3.9**, P.32²²

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, **CHAPTER 3**, P.32²³

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, **CHAPTER 3 EQ:3.10**, P.32²⁴

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, **CHAPTER 3**, P.32²⁵

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, **CHAPTER 3 EQ:3.11**, P.32²⁶

حيث أن متغيرات النموذج هي: $\theta_1, \theta_2, \theta_3, \dots, \theta_k$

يجب أن تكون نتيجة التصنيف عبارة عن تدرج، لكن توسيع قيمة الخرج لا يمكن اعتبارها كدرج، لأنه بسبب هذا التوسيع للمصنف المنطقي باستخدام النموذج التفرعي. من أجل كل فئة، ينتج ز خرج قيمة احتمالية محسوبة، والتي تدل على احتمالية أن كائن البيانات يتم تقسيمه إلى فئات. لقد حقق التصنيف مع الفة المقابلة لقيمة الاحتمال العظمى²⁷.

2.4 - مرحلة الاختبار:

نظام العمل يستخدم جهاز التصوير (*Kinect*) الموجود على رأس روبوت له عجلات من أجل تتبع تعابير الوجه في الصور أولاً، وبعدها يستدعي خوارزمية التعرف على تعابير المشاعر في الوجه من أجل استخراج الميزات والتعرف على المشاعر، والتي تم تصميمها باستخدام ماتلاب (*MATLAB*)²⁸.



Samples of the JAFFE facial expression library

الشكل رقم (3): عينات من مكتبة (JAFFE) لتعابير الوجه²⁹.

إن قاعدة بيانات (JAFFE) لتعابير الوجه هي بالأصل يابانية تستخدم من أجل تدريب العينات، والتي تحتوي على 213 صورة لتعابيرات الوجه، لعشر نساء من اليابان، يوجد بها سبع أنواع من المشاعر³⁰.

2.4.1Fine-Tune Effect on Performance of Recognition:

تم تصميم تصور الخصائص الأساسية للأوزان التي تم تعلمها عن طريق (sparse autoencoder network)، وتم تحديد عدد عقد الخلايا العصبية في الطبقة المخفية على 140 للحصول على صورة أولية لتصور الخصائص. يتم عرض مصفوفة وزن الخصائص المرئية على النحو التالي:

الشكلان (3.7 & 3.8) يقدمان مصفوفة مميزة لصور الأوزان قبل وبعد ال (fine – tuning)، نرى أن الميزات هنا تكون ذاتية التعلم يمكن ملاحظة أن الميزات المكتسبة ذاتياً من قبل الشبكة الكلية تبدو أكثر تعقيداً بعد ال (fine – tuning) لضمان دقة التعرف العالمية.

وفي الشكل (3.9) نرى منحنى العلاقة بين الأنواع السبعة لمعدل التعرف على الوجه ومعامل التباين. وفقاً لمقارنة الشكل (3.10) مع الشكل (3.11)، نستنتج أن ال (fine – tuning) يمكن أن يجعل دالة التكلفة الإجمالية تتقارب بشكل أسرع، وفي الاختبار الفعلي (cost function) 182 مرة عندما يتقاربون ويتوقفون عن التدريب³¹.

Fig. 3.7 The weights visualization of the underlying characteristics



Fig. 3.8 The fine-tuning the weights visualization of the underlying characteristics

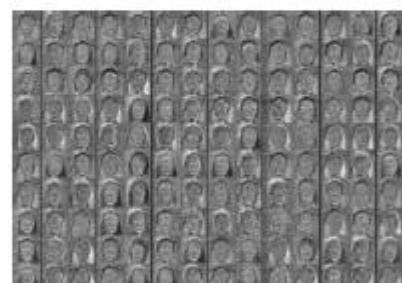


Fig. 3.9 The influence of sparse parameter and fine-tuning the weights on the rate of facial emotion recognition

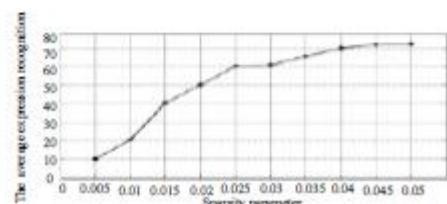


Fig. 3.10 The convergence of overall cost function before fine-tuning

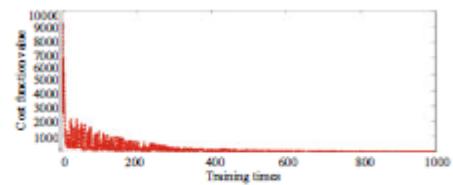


Fig. 3.11 The convergence of overall cost function after fine-tuning

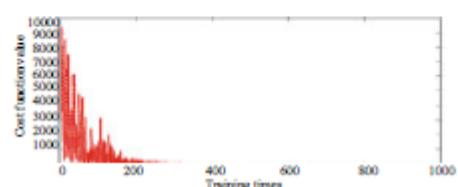


Fig. 3.12 Weights visualizations for the hidden layer node number is 50

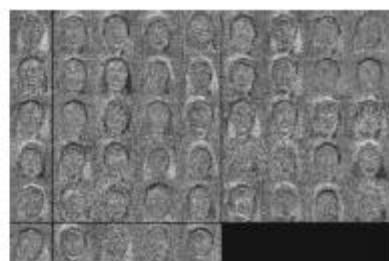


Fig. 3.13 Weights visualizations for the hidden layer node number is 200

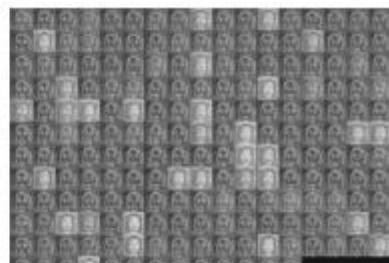
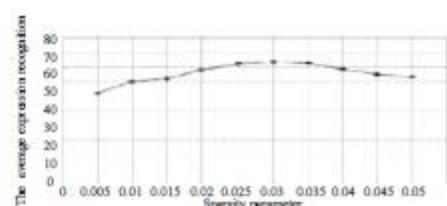


Fig. 3.14 The influence of the number of hidden layer nodes on the expression recognition rate



2.4.2 –The Number of Hidden Layer Node's Effect on Performance of Recognition:

نقوم بتصميم تصور الخصائص الأساسية للأوزان في شبكة التشفير التقاني المتفرقة وتغيير عدد عقدة الخلايا العصبية في الطبقة المخفية لمراقبة التغييرات في صور ميزة مصفوفة الوزن. يتم عرض مصفوفة وزن الخصائص المرئية على النحو التالي:

يمكنا أن نرى من الشكلين (3.12 & 3.13) أن زيادة عدد عقد الطبقات المخفية من الممكن أن يزيد معدل التعرف على المشاعر، ومع ذلك، فإن عدد كبير من عقد الطبقات المخفية لا يكون له تأثير بتحسين معدل التعرف على المشاعر. وهي وبالتالي سوف تسبب ما يسمى بال (*Overfitting*) للشبكة ولن يؤدي إلى استخراج ميزات تعابير الوجه. بعد ال (*fine – tuning*) لمصفوفة الوزن، معدل التعرف سوف يتحسن إلى حد معين، ويتم تعويض تأثير تغيير أرقان الطبقة المخفية، كما هو موضح في الشكل (3.14).³³

Emotion	Softmax regression (%)	SRDSAN (%)
Natural	86.667	100.00
Happy	80.000	93.333
Sad	63.333	100.00
Fear	63.333	76.667
Angry	83.333	100.00
Disgust	66.667	93.333
Surprise	70.000	100.00
Average	73.333	94.761

Comparison of emotion recognition experimental results

الجدول رقم (2): مقارنة النتائج التجريبية للتعرف على المشاعر.³⁴

2.4.3- Recognition Rate:

للحصول من دقة التعرف على المشاعر، تجارب المحاكاة تم إكمالها باستعمال (*MATLAB*)، ومع إدراج النتيجة بالجدول رقم (2).

وفقاً للجدول، فإن متوسط الدقة ل (*Softmax Regression*) هو 73.33% في الاختبار النهائي. ومع ذلك، إذا استخدمنا (*deep autoencoder network*) من أجل تدريب ال (*unlabeled training data*) أولًا، ومن ثم تدريب نموذج ال (*SR*)، نجد أن عدد عمليات تقارب التكرار هو 181 فقط، والدقة هي 94.76%.

من خلال دمج تراجع ال (*Softmax*) في التعلم العميق، يمكن ملاحظة الميزات التي يتم اكتسابها ذاتياً من قبل الشبكة الكلية أنها تكون أكثر تعقيداً بعد ال (*fine – tuning*), وال (*fine – tuning*) يجعل دالة التكلفة الإجمالية

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 3 , P.36³³

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 3 TABLE 3.2 , P.37³⁴

(Overall Cost Function) تتقرب بشكل أسرع، والذي يتغلب بدوره على مشاكل الانتشار المتدرج والقيم المحلية القصوى. غير ذلك، يوضح هذا أن الخصائص التي تعلمتها (Sparse Autoencoder Network) ذاتية (Sparse Autoencoder Network) ذاتية التعلم لدينا هي أكثر تمثيلاً من خصائص بيانات الإدخال الأصلية، وهو الاختلاف النموذجي بين طرق التدريب التقليدية وطرق التدريب على التعلم العميق. وأيضاً، يوضح هذا أنه بالإضافة إلى ذلك، يتغير شكل وقيم الأجزاء الثلاثة الرئيسية (منطقة الاهتمام) مثل الحاجبين والعينين والفم التي بدورها قد تعكس تغييرات دقيقة في ميزات تعبيرات الوجه.³⁵

2.5: الملخص:

يقترح استخدام (SRDSAN) من أجل التعرف على مشاعر الوجه، حيث يتم استخدام أداة تشغيل تلقائي وال (SR) وتم توظيفهم في التعلم العميق للتعرف على تعبيرات الوجه. من أجل التحقق من دقة الاقتراح، التجارب الأولية تم إكمالها باستخدام (MATLAB) باستخدام قاعدة تعبيرات وجه يابانية قياسية (JAFFE). متوسط التعرف على المشاعر وصل إلى 77.333% باستخدام نموذج ال (SR)، لكن إذا استخدمنا (SRDSAN) المقترن، فإن متوسط التعرف على المشاعر يرتفع إلى 94.761% وفقاً للنتائج التجريبية، من خلال دمج تراجع ال (Softmax) في التعلم العميق، يمكن ملاحظة أن الميزات المكتسبة ذاتياً من قبل الشبكة الكلية تبدو أكثر تعقيداً بعد ال (fine – tuning)، كما أن ال (fine – tuning) يجعل وظيفة التكلفة الإجمالية تتقرب بشكل أسرع، مما يتغلب على القيم المحلية القصوى، والتدرج ومشاكل الانتشار.

3. AdaBoost-KNN with Direct Optimization for Dynamic Emotion Recognition

.3 مع التحسين المباشر للتعرف على المشاعر الديناميكية.

(AdaBoost – KNN) باستخدام اختيار ميزات تكيفية مع التحسين المباشر مقترنة من أجل التعرف على المشاعر الديناميكية في التفاعل بين الإنسان والحاسب، في حين أن يتم التعرف بالزمن الحقيقي على المشاعر الديناميكية بناءً على تعبيرات الوجه. يمكن أن يجعل الروبوتات قادرة على فهم المشاعر الديناميكية البشرية، بطريقة تتحقق التفاعل بين الإنسان والروبوت بطريقة سلسة. يعتمد تصنيف المشاعر على (AdaBoost – KNN)، الذي يبني سلسلة مصنفات (KNN) يضبط (AdaBoost – KNN) أوزان البيانات بطريقة تكرارية، وغير ذلك، ال

(*global optimal parameters*) يتم تقريبها بالتحسين المباشر حتى يصل معدل التعرف إلى قيمته القصوى³⁶.

3.1 AdaBoost-KNN Based Emotion Recognition:

إن ال (*AdaBoost – KNN*) هو عبارة عن مصنف متكامل، يتكون من مجموعة من ال (*kNN*) عبارة (*test data & train data*), حيث أن كل من ال (*test data, train data, train label, k*) عن متغيرات ذاتية مستقلة. ال (*Train Label*) هي التسمية الخاصة بعينة بيانات التدريب $[19]$. $k = 1, 2, \dots, N$.

تمثل عدد عينات k الأكثر تشابهًا مع عينات الاختبار الموجودة في فضاء الميزة. في بيانات ميزات الوجه المعطاة، نتيجة كل تصنیف (*KNN*) يحدد نتیجة التصنیف المثالي. الخطوات من أجل إنشاء المصنف تكون كالتالي:

الخطوة الأولى: تهيئة أوزان بيانات التدريب.

الخطوة الثانية: استخدام مجموعة بيانات التدريب مع معامل الوزن ($D(m)$) لإنشاء المصنف الضعيف ($Gm(x)$), وحساب معدل خطأ التصنیف ل($Gm(x)$ على بيانات التدريب، وحساب معالم (x).

الخطوة الثالثة: تحديث توزيع قيمة تمركز البيانات، وبناء المصنف النهائي وفقاً (.coefficient combination basic classifier).

الخطوة الرابعة: تدريب المصنف باستخدام 70% من بيانات (*JAFFE*) للتدريب، و30% من البيانات تستخدم للاختبار. ونتيجة، تم تدريب 7 مصنفات.

الخطوة الخامسة: استخدام (*AdaBoost – KNN*) من أجل تصنیف المشاعر السبعة الأساسية. وتكون عملية اتخاذ القرار كما يلي³⁷:

$$D_1 = (\omega_{11}, \dots, \omega_{1i}, \dots, \omega_{1N})$$

$$\omega_{1i} = \frac{1}{N}, i = 1, 2, \dots, N$$

38

حيث α_{1N} هي الوزن الابتدائي لعينة التدريب. كل عينة تدريب يتم إعطاؤها نفس الوزن في البداية. N هو عدد عينات التدريب، و($D(m)$) يتم استخدامها لتخزين مصفوفة M تمثل أول رقم من التكرارات $\omega_{li} = \frac{1}{N}, i = 1, 2, \dots, M$ وعدد المصنفات الضعيفة. باستخدام مجموعة بيانات التدريب مع معامل الوزن ($D(m)$) لإنشاء المصنف

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 , P.41³⁶

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 , P.45³⁷

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 EQ.4.8+EQ4.9 , P.45³⁸

الضعيف($G_m(x)$ ، وحساب نعدل خطأ التصنيف e_m ل($G_m(x)$) على مجموعة بيانات التدريب، وحساب معامل ω_m ³⁹. $G_m(x)$

$$e_m = \sum_{i=1}^N \omega_{mi} I(G_m(x_i) \neq y_i)$$

$$\alpha_m = \frac{1}{2} \log \frac{1 - e_m}{e_m}$$

40

وبعدها قم بتحديث توزيع قيمة تمركز البيانات.

$$\omega_{m+1,i} = \frac{\omega_{mi}}{Z_m} \exp(-\alpha_m y_i G_m(x_i)) \quad i = 1, 2, \dots, N$$

$$D_{m+1} = (\omega_{m+1,1}, \omega_{m+1,2}, \dots, \omega_{m+1,N})$$

$$Z_m = \sum_{i=1}^N \omega_{mi} \exp(-\alpha_m y_i G_m(x_i))$$

41

تكون أهمية Z_m هي في جعل مجموع عامل الترجيح تساوي الواحد، بحيث يكون المتجه D متوجه توزيع احتمالات. وبعدها نقوم بدمج كل مصنف ضعيف للحصول على المصنف النهائي. α_M متزداد مع انخفاض e_m ، بما معناه أنه كل ما انخفض معدل تصنيف الخطأ، كلما زاد دور المصنف في المصنف النهائي⁴².

$$G(x) = sign(f(x)) = sign(\sum_{m=1}^M (\alpha_m G(x)))$$

43

كل مصنف ضعيف هو مصنف (KNN) منفصل^[20]. خطوات تنفيذ (KNN) كالتالي:

الخطوة الأولى: قم بحساب المسافة الإقليدية بين النقاط في مجموعة بيانات الفئة المعروفة، ونقاط عينة الاختبار.

الخطوة الثانية: قم بترتيب نقاط عينة التدريب بشكل متزايد بالنسبة للمسافة.

الخطوة الثالثة: قم باختيار k نقاط عينة تدريب بأقل مسافة من النقطة الحالية.

الخطوة الرابعة: قم بتحديد تكرار حدوث فئة النقاط k ، وقم بإرجاع الفئة ذات أعلى تردد نقاط k كتصنيف تبؤ لنقطة الاختبار⁴⁴.

المقياس الإقليدي (المسافة الإقليدية) هو مسافة شائعة الاستخدام^[21]. تشير إلى المسافة الحقيقية بين نقطتين، أو طول متوجه طبيعي في فضاء أبعاد. في الفضاء الإقليدي الذي بعده هو N . المتوجه x يمكن تمثيله بالصيغة: $x = (x_1, x_2, \dots, x_n)$ وهو الرقم الحقيقي، الذي يدعى بالإحداثية i_{th} ، لـ x المسافة $p(A, B)$ بين نقطتين $A = (a_1, a_2, \dots, a_n)$ و $B = (b_1, b_2, \dots, b_n)$ تكون على النحو التالي⁴⁵:

$$\rho(A, B) = \sqrt{\sum (a_i - b_i)^2} \quad (i = 1, 2, \dots, n) \quad 46$$

الطول الطبيعي للمتوجه $x = (x_1, x_2, \dots, x_n)$ يعرف على النحو التالي:

$$|x| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2} \quad 47$$

إنّ عدد أقرب جيران هو k ، ومن الممكن أن يعكس مدى تعقيد نموذج بأكمله. كلما كان k أصغر، كلما ارتفع تعقيد النموذج، ومن السهل أن يصبح (overfitting). في التطبيق يكون هناك ازدياد للفرق بين بيانات العينة وتم غالباً تم استخدام k أصغر. عموماً، تحديد الائمثل k هو عن طريق الـ (cross validation)⁴⁸.

3.2 AdaBoost-KNN with Direct Optimization for Dynamic Emotion Recognition

طريقة التحسين المباشر (DA) تجد مجموعة من المتغيرات المثالية في فضاء المتغيرات عن طريق البحث المباشر، وهي أحد الطرق العددية. في الأساس، خلال عملية التدريب الكاملة، سيظهر المصنف الضعيف بعض أخطاء التصنيف، مهما كان عدد جولات التدريب. ومع ذلك، من المرجح أن يتقارب إطار (AdaBoost) بأكمله بسرعة. لذلك، من المهم تحديد الأساس من أجل تحديد نهاية التكرار من أجل البحث عن المتغيرات المثالية⁴⁹.

بعد كل جولة تدريب، يقوم الـ (AdaBoost) بضبط وزن العينات، ونتيجة هذا التعديل هي أن مزيداً من العينات يتم تصنيفها لاحقاً بشكل خاطئ، وأن وزن العينات يصبح أعلى. بهذه الطريقة، فإن المصنف الضعيف سيحقق خطأ تصنيفي ضعيف من خلال تصنيف العينات ذات الوزن المرتفع بشكل صحيح. على الرغم من ذلك، فإن مصنف ضعيف يمكن أن

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 , P.46⁴⁴

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 , P.46⁴⁵

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 EQ.4.16 , P.47⁴⁶

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 EQ.4.17 , P.47⁴⁷

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 , P.47⁴⁸

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 , P.47⁴⁹

يسbib حدوث أخطاء في التصنيف من تلقاء نفسه، جميع العينات التي تم تصنيفها بشكل خاطئ ستكون ذات وزن منخفض، والخرج النهائي لإطار الـ *(AdaBoost)* سوف يكون "موازن" بواسطة المصنف الضعيف ذو الوزن المرتفع المصنف بشكل صحيح مسبقاً⁵⁰.

قم بتسجيل عدد تكرارات ومعدل أخطاء التعرف على المشاعر $(e(i))$ بعد كل جولة من التكرار، ثم أكمل التكرار التالي عندما تقوم الخوارزمية بضبط وزن العينات والمصنف باستمرار. عندما تنتهي عملية التكرار، قم بتسجيل عدد التكرارات $(1 + i)$ ومعدل خطأ تعرف المشاعر $(e(i))$. إذا كان $(e(i + 1) > e(i))$ ، يسجل $(e(i + 1) \& e(i + 1))$

من خلال تالي التكرارات، حتى يتوقف معدل خطأ خوارزمية الـ *(AdaBoost – KNN)* عن الانخفاض، تم الوصول إلى شرط التقارب. وأخيراً، خارج الحلقة، المتغيرات $(1 \& e(i + 1))$ التي تم الحصول عليها، تكون هي المتغيرات المثالية القابلة للتطبيق على النموذج الحالي.⁵¹

لذلك، عند تصميم شرط التقارب الخاص بالتعرف على المشاعر، نعتمد طريقة تسجيل معدل التعرف على الذروة، والخطوات كالتالي:

الخطوة الأولى: قم بحساب معدل التعرف على مشاعر الوجه. $(e(i - 1))$.

الخطوة الثانية: قم بزيادة قيمة k وابدا بالجولة التالية من الحساب.

الخطوة الثالثة: قم بحساب معدل التعرف على مشاعر الوجه. $(e(i + 1) - 1)$.

الخطوة الرابعة: قم بتسجيل معدل التعرف الذي تم الحصول عليه من خلال جولتين من الحساب. إذا كان معدل التعرف قد بدأ بالتناقض، أو قف الحلقة، وسجل قيمة k ومعدل التعرف في هذه اللحظة، أي، القيمة الأمثل للمتغيرين. غير ذلك، قم بزيادة k واستمر في الخوارزمية.

الخطوات لتحديد عدد التكرارات التجريبية، يكونوا كالتالي:

الخطوة 1: قم بحساب معدل الخطأ للتعرف على مشاعر الوجه.

الخطوة 2: قم بزيادة (i) وابدا الجولة التالية من الحساب.

الخطوة 3: قم بحساب معدل الخطأ من التعرف على مشاعر الوجه $(e(i + 1))$.

الخطوة 4: قم بتسجيل معدل الخطأ للتعرف على مشاعر الوجه التي تم الحصول عليها من جولتين من الحساب. إذا كان المعدل التعرف يبدأ بالإزدياد، أو قف الحلقة، وقم بتسجيل قيمة (i) ومعدل التعرف في هذه اللحظة، أي، القيمة الأمثل للمتغيرين، غير ذلك، قم بزيادة (i) وتتابع تنفيذ الخوارزمية⁵².

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 , P.47⁵⁰

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 , P.47⁵¹

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 , P.47+48⁵²

3.3 – التجارب:

تم اختيار (JAFFE) كقاعدة بيانات للتحقق من فاعلية الخوارزمية المقترنة. في الوقت نفسه، تجربة متغير التحكم تم تصميمها لاستكشاف تأثير قيمة كل متغير في خوارزمية (AFS – AdaBoost – KNN – DO) على تأثير التعرف. وأخيراً خوارزمية التعرف على المشاعر الديناميكية للوجه اكتملت على نظام تفاعل الروبوت الاجتماعي العاطفي، والنتيجة يتم تحليلها على أساس التطبيق.⁵³

3.3.1 – البيئة التجريبية واختيار البيانات:

قاعدة البيانات التي تم اعتمادها من أجل التعرف على مشاعر الوجه هي (JAFFE public database).^[22]

هناك من صورتان لأربع صور لكل شخص. 70% من الصور تم اختيارها للتدريب، بينما الـ30% الأخرى من أجل الاختبار. مثال عن الصور من قاعدة البيانات:⁵⁴



55

Partial sample image of the database

3.3.2 – المحاكاة والتحليل:

(L, D, R) هي ثلاثة متغيرات تم تضمينها في ال (LRS – based AFS)، حيث (L) هي عدد الميزات المضافة كل دورة، و(R) عدد الميزات التي تمت إزالتها في كل دورة، و(D) هو العدد الإجمالي للميزات التي تم تركها من قبل الدورة الأخيرة. عندما تأخذ المتغيرات (L, D, R) قيمًا مختلفة، فإن معدل التعرف على المشاعر الديناميكية مع كل متغير كما هو موضح أدناه⁵⁶:

عندما تكون قيمة (R = 124, D = 540, L = 180) تتغير قيمة (L)، وعندما (D = 540, L = 180)، تتغير قيمة (R)، التغيير في معدل التعرف على المشاعر الديناميكية يتم عرضه كالتالي:

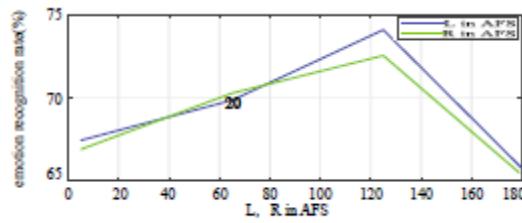
Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 , P.48⁵³

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 , P.48⁵⁴

Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 Fig4.2 , P.49⁵⁵

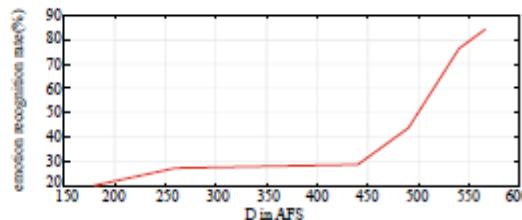
Emotion Recognition and understanding for Emotional Human-Robot Interaction Systems, CHAPTER 4 Fig4.2 , P.49⁵⁶

Dynamic facial emotion recognition rate varies with L, R



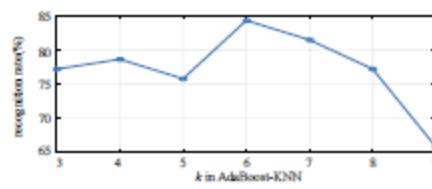
عندما تكون قيمة $(L = 125, R = 124)$ ، فإن عدد مجموعات الميزات المختارة أخيراً تتغير، التغير في معدل التعرف على المشاعر الديناميكية يتم عرضه كالتالي:

Dynamic facial emotion recognition error rates varies with D



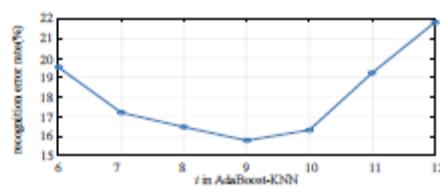
في تجربة حكم التقارب لمعدل خطأ التعرف، مع زيادة k ، ترتفع دقة التعرف في البداية ثم تنخفض، كالتالي:

Dynamic facial emotion recognition rate as a function of k



مع زيادة عدد التكرارات (1)، يتناقص خطأ التعرف أولًا ثم يزداد. كالتالي:

Dynamic facial emotion recognition error rates as a function of i



من أجل مناقشة تأثير خوارزمية تحسين (*AdaBoost – KNN*) على معدل التعرف على المشاعر الديناميكية، قمنا بتصميم تجربة مقارنة. معدلات التعرف، والبيانات ومعاملات (*kappa*) للمجموعات الثلاثة من التجارب، توضح في الجدول التالي:

Dynamic facial emotion recognition rate for selected algorithms

Algorithm	Recognition rate (%)	Variance	Kappa coefficient
AdaBoost-KNN	81.42	7.38	0.78
AFS-AdaBoost-KNN	88.57	3.41	0.86
AdaBoost-KNN-DO	91.42	5.06	0.89
AFS-AdaBoost-KNN-DO	94.28	0.67	0.93

إن الـ **لتنفيذ خوارزمية (Adaboost – KNN) (confusion matrix)** تكون كالتالي:

Confusion matrix
of emotion recognition by
using AdaBoost-KNN

	AN.	DI.	FE.	HA.	NE.	SA.	SU.
AN.	.80	.10	.10	0	0	0	0
DI.	0	.90	.10	0	0	0	0
FE.	0	0	.80	0	0	.10	.10
HA.	.30	0	0	.70	0	0	0
NE.	0	0	0	0	1	0	0
SA.	0	0	.10	0	.10	.70	.10
SU.	0	0	0	.10	.10	0	.80

عند تنفيذ خوارزمية (AFS – AdaBoost – KNN) (confusion matrix)، فتكون الـ (AFS – AdaBoost – KNN)، كالتالي:

Confusion matrix
of emotion recognition using
AFS-AdaBoost-KNN

	AN.	DI.	FE.	HA.	NE.	SA.	SU.
AN.	.90	.10	0	0	0	0	0
DI.	0	1	0	0	0	0	0
FE.	0	0	.90	0	0	.10	0
HA.	.30	0	0	.70	0	0	0
NE.	0	0	0	0	1	0	0
SA.	0	0	0	0	0	.90	.10
SU.	0	0	0	.10	.10	0	.80

وعند اعتماد خوارزمية التحسين المباشر ل (*AdaBoost – KNN*), والتي تدعى ب (*AdaBoost – KNN – DO* (confusion matrix)، فتكون الـ (*KNN – DO*)، كالتالي:

Confusion matrix
of emotion recognition by
using AdaBoost-KNN-DO

	AN.	DI.	FE.	HA.	NE.	SA.	SU.
AN.	1	0	0	0	0	0	0
DI.	0	.90	0	0	0	.10	0
FE.	0	0	.90	0	0	.10	0
HA.	0	0	0	.90	.10	0	0
NE.	0	0	0	0	1	0	0
SA.	0	0	0	0	.10	.90	0
SU.	0	0	0	0	.20	0	.80

عندما يتم اكتمال كل من (AFS) وخوارزمية التحسين المباشر ل (AdaBoost – KNN – DO) التي تدعى بـ (confusion matrix) (AdaBoost – KNN – DO)، تكون:

Confusion matrix
of recognition using
AFS-AdaBoost-KNN-DO

	AN.	DI.	FE.	HA.	NE.	SA.	SU.
AN.	1	0	0	0	0	0	0
DI.	0	1	0	0	0	0	0
FE.	0	0	.90	0	0	.10	0
HA.	.20	0	0	.80	0	0	0
NE.	0	0	0	0	1	0	0
SA.	0	0	0	0	.10	.90	0
SU.	0	0	0	0	0	0	1

معامل (Kappa) هو مقياس دقة التصنيف. عموماً، تكون قيمة معامل (Kappa) بين ال (0 & 1)، ويمكن تقسيمها إلى خمس مجموعات، كالتالي:

0 → 0.20	اتفاقاً طفيفاً
0.21 → 0.40	اتفاق عادل
0.41 → 0.60	اتفاق معتدل
0.61 → 0.80	اتفاق عالي
0.81 → 1	اتفاق مثالي تقربياً

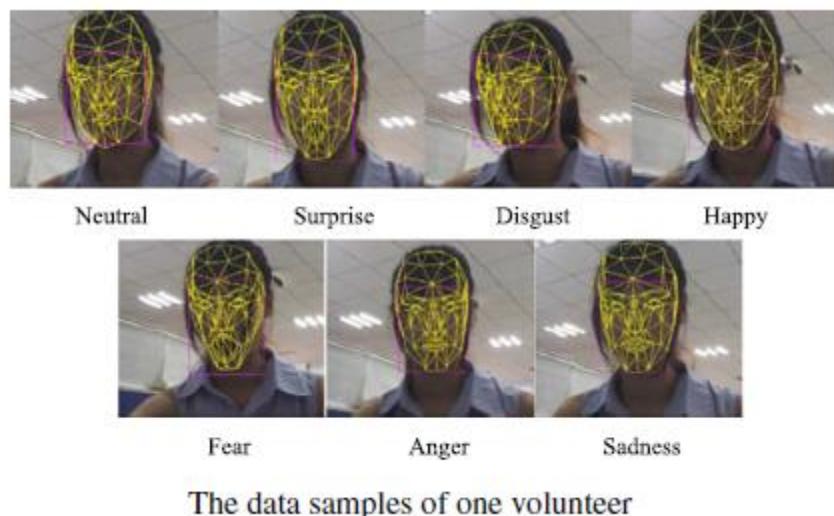
نُظِّمَ النَّتْيُودُ أَنَّ نَتْيُودَ التَّعْرِفَ لِلطَّرِيقَةِ (AdaBoost – KNN) تَتَلَقَّى اتِّفَاقاً عَالِيًّا.⁵⁷

Algorithm	Algorithm Recognition rate (%)	Variance
GM – PSO – SVM[23]	89.00	\
LSFA – SVM [24]	85.95	3.82
ODFA – AdaBoost[25]	88.78	\
Chaotic FA – KNN[26]	91.45	\
AFS – AdaBoost – KNN – DO	94.28	0.67

بالمقارنة مع طرق التعرف على مشاعر الوجه الأخرى، فإن الجدول السابق يعرض معدل التعرف في الخوارزميات المختلفة⁵⁸.

3.3.3 تجارب التطبيق الأولية على النظام:

يتم التعرف على المشاعر الديناميكية في التطبيق الأولي. تمت دعوة عشرة متطوعين تتراوح أعمارهم بين 18 و28 عاماً لبناء البيانات التجريبية. يعبر كل متطوع عن سبع مشاعر أساسية: السعادة، والغضب، والحزن، والاشمئاز، والمفاجأة، والخوف، والحياء. يتم جمع عينات البيانات لأحد المتطوعين، كالتالي:



وفقاً لتعبير الوجه الديناميكي في الزمن الحقيقي بالنسبة للبشر، يتم التعرف على 70 شعور ديناميكي لـ 10 متطوعين باستخدام مطابقة نقطة الميزة ومطابقة نقاط الميزة الديناميكية القائمة على (*Candide* – 3). تكون ال خاصة بالتعرف على المشاعر كالتالي: (*confusion matrix*)

Confusion matrix of recognition results obtained by using AFS-AdaBoost-KNN-DO

	AN.	DI.	FE.	HA.	NE.	SA.	SU.
AN.	.80	.10	.10	0	0	0	0
DI.	0	.80	.10	0	0	.10	0
FE.	0	0	.70	0	0	0	.30
HA.	0	0	.10	.90	0	0	0
NE.	0	0	.10	0	.90	0	0
SA.	.10	0	0	0	0	.90	0
SU.	0	0	.30	0	0	0	.70

وفقاً لتحليلات النتائج لمتوسط معدل التعرف والـ (*confusion matrix*) يبلغ متوسط معدل التعرف 81.42% ومعامل (*KAPPA*) 0.78 باستخدام الخوارزمية المقترنة. نتائج التعرف الناتجة عن طريقة (*AdaBoost*) – (*KNN* – *DO*) تتوافق إلى حد كبير مع عينة الاختبار. بينما كان متوسط معدل التعرف 72.34% عندما لم يتم استخدام (*AFS*). نرى من خلال التجربة التطبيقية لنظام التفاعل بين الإنسان والروبوت أن نظام التعرف لديه معدل تمييز عالٍ لخمس مشاعر: الحيادية، والحزن، والسعادة، والاشمئزاز، والغضب. لكن الخوف والمفاجأة لا يتم التعرف عليهما بسهولة. تحتوي جميع تعبيرات الخوف والعاطفة المفاجئة على خصائص متشابهة مثل التحديق الدائري بالعينين، لذلك يصعب التمييز بينهما⁵⁹.

النتائج أظهرت أن الخوارزمية المقترنة للتعرف على المشاعر الديناميكية للوجه في هذا القسم من البحث لديها أداء قوي في الزمن الحقيقي. وغير ذلك، علاوة على ذلك، يأخذ الاقتراح في الاعتبار مساهمة النقاط المميزة المختلفة في التعرف على تعبيرات الوجه، والتي يمكن أن تكشف تباين نقاط السمات الرئيسية في تعبيرات الوجه، بحيث تم تحسين معدل التعرف على المشاعر بنسبة 9.08%.

3.4 – الملخص:

خوارزمية (*AFS* – *AdaBoost* – *KNN* – *DO*) تم اقتراحتها من أجل التعرف على مشاعر الوجه الديناميكية على أساس مصنف (*KNN*). فإن (*AdaBoost*) يمكنها ضبط أوزان العينة والمصنف. بينما، خوارزمية (*AFS*) اعتمدت طريقة اختيار الميزة. وأيضاً، تم اعتماد التحسين المباشر لتحديد متغيرات النموذج تلقائياً وتحقيق تحسين للنموذج. تظهر النتائج التجريبية أن معدل التعرف على الاقتراح أعلى من معدل كل من كل من الخوارزميات التالية: (*AdaBoost* – *KNN*, *AFS* – *AdaBoost KNN*, *AdaBoost* – *KNN* – *DO*) طريق التعرف التقليدية الأخرى، مثل: (*AdaBoost*, *KNN*, *SVM*) وغيرها الكثير⁶⁰.

أصبح التعرف على المشاعر وفهم المعلومات أكثر انتشاراً في التفاعل بين الإنسان والروبوت، وموضوع أن الروبوتات اليوم تتعرف على المشاعر البشرية وتفهمها وتتكيف معها من خلال السلوك من الممكن أن يكون من عناوين موضوعات بحثية مثيرة للاهتمام⁶¹.

المراجع:

1. S.E. Kitayama, H.R.E. Markus, Emotion and Culture: Empirical Studies of Mutual Influence (American Psychological Association, 1994)
2. B. Parkinson, A.H. Fischer, A.S.R. Manstead, Emotion in Social Relations: Cultural, Group, and Interpersonal Processes (Psychology Press, 2005)
3. R. Ekman, What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS) (Oxford University Press, USA, 1997)
4. D. Schuller, B.W. Schuller, The age of artificial emotional intelligence. Computer 51(9), 38–46 (2018)
5. F. Li, J. Feng, M. Fu, A natural human-robot interactionmethod in virtual roaming, in Proceedings of the 2019 15th International Conference on Computational Intelligence and Security (CIS) (2019)
6. A. Vinciarelli, M. Pantic, H. Boulard, Social signal processing: survey of an emerging domain. Image Vis. Comput. 27(12), 1743–1759 (2009)
7. J.C. Gómez, A.G. Serrano, P. Martínez, Intentional processing as a key for rational behaviour through natural interaction. Interact. Comput. 18(6), 1419 1446 (2006)
8. J. Fan, M. Campbell, B. Kingsbury, Artificial intelligence research at IBM. IBM J. Res. Dev. 55(5), 16: 1–16: 4 (2011)
9. X.Q. Zheng, M. Shiomi, T. Minato, H. Ishiguro, What kinds of robot's touch will match expressed emotions. IEEE Robot. Autom. Lett. 5(1), 127–134 (2019)
10. P. Christopher, L. Johnson, D.A. Carnegie, Mobile robot navigation modulated by artificial emotions. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) 40(2), 469–480 (2009)
11. M. Wu, Z.T. Liu, L.F. Chen, AffectiveComputing and Emotional Robot System (Beijing Science Press, Peking, 2018)
12. L. Zhang, D. Tjondronegoro, Facial expression recognition using facial movement features. IEEE Trans. Affect. Comput. 2(4), 219 229 (2011)
13. C.K. Hsieh, S.H. Lai, Y.C. Chen, An optical flow-based approach to robust face recognition under expression variations. IEEE Trans. Image Process. 19(1), 233–240 (2010)
14. Z. Li, D. Gong, X. Li, D. Tao, Aging face recognition: a hierarchical learning model based on local patterns selection. IEEE Trans. Image Process. 25(5), 2146–2154 (2016)
15. M.A.A. Dewana, E. Grangerb, G.-L. Marcialisc, R. Sabourinb, F. Rolic, Adaptive appearance model tracking for still-to-video face recognition. Pattern Recognit. 49, 129–151 (2016)
16. E. Vezzetti, F. Marcolin, G. Fracastoro, 3D face recognition: an automatic strategy based on geometrical descriptors and landmarks. Robot. Auton. Syst. 62(12), 1768–1776 (2014)
17. L. Xu, W.W. Liu, K. Tsujino, C.W. Lu, A facial recognition method based on 3-D images analysis for intuitive human-system interaction, in Proceedings of International Joint Conference on

Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA), Aizuwakamatsu, Japan, pp. 371–377 (2013)

18. B.J.T. Fernandes, G.D.C. Cavalcanti, T.I. Ren, Face recognition with an improved interval type-2 fuzzy logic sugeno integral and modular neural networks. *IEEE Trans. Syst. Man Cybern.-Part A: Syst. Hum.* 41(5), 1001–1012 (2011)
19. J. Maillo, F. Ramírez, I. Triguero, F. Herrera, K-NN-IS: An iterative spark based design of the k-nearest neighbors classifier for big data. *Knowl.-Based Syst.* 117, 3–15 (2017)
20. X.L. Li, Q.M. Lu, Y.S. Dong, A manifold regularized set-covering method for data partitioning. *IEEE Trans. Neural Netw. Learn. Syst.* 29(5), 1760–1773 (2018)
21. Y. Wen, L. He, P. Shi, Face recognition using difference vector plus KPCA. *Dig. Signal Process.* 22(1), 140–146 (2012)
22. The Japanese Female Facial Expression (JAFFE) Database, <http://www.kasrl.org/jaff.html>
23. Y. Zhang, L. Zhang, S.C. Neoh, K. Mistry, A. Hossain, Intelligent affect regression for bodily expressions using hybrid particle swarm optimization and adaptive ensembles. *Expert Syst. Appl.* 42(22), 8678–8697 (2015)
24. O.P. Verma, D. Aggarwal, T. Patodi, Opposition and Dimensional based modified firefly algorithm. *Expert Syst. Appl.* 44, 168–176 (2016)
25. Y. Guo, Y. Liu, A. Oerlemans, Deep learning for visual understanding: a review. *Neurocomputing* 187, 27–48 (2016)
26. W. Zhang, Y. Zhang, L. Ma, J.W. Guan, S.J. Gong, Multimodal learning for facial expression recognition. *Pattern Recogn.* 48(10), 3191–3202 (2015)

القسم الثاني

الدراسة المرجعية

شركات تعتمد على فكرة مشروعنا

بشكل عام التعرف على مشاعر الإنسان له فوائد عديدة في فهم ردود أفعال البشر في كثير من الأماكن، وبسبب كون فكرة مشاعر الإنسان ذات قيمة كبيرة، فقد صدرت عنها تقنية التعرف على المشاعر.

وتقنية التعرف على المشاعر هي نوع من أنواع الذكاء الاصطناعي المتعلقة بالتعرف على الوجوه، والتي تحاول التعرف على شعور الإنسان بناءً على تعابير وجهه وبباقي أجزاء جسمه، بما في ذلك معدل ضربات القلب ونشاط الدماغ. إن هذه البرمجية بإمكانها أيضاً تتبع حرك العين للتعرف إلى على أكثر شيء يعطيه الإنسان اهتمامه. وهي تتطلب تقنيات التعلم العميق من أجل تحسين جمعها للمعلومات⁶².

ومن حالات استخدام الذكاء الصنعي للتعرف على المشاعر:

في مجال الرعاية الصحية، وفي مجال التسويق، وفي مجال الصناعة، وغيرها الكثير، ومن هذه الحاجة لنظم التعرف على المشاعر انطلقت العديد من الشركات المختصة بهذه التقنية.

1. *Affectiva*:

:) Affectiva



وهي شركة برمجة، تبني نظم ذكاء اصطناعي تفهم مشاعر البشر، وتفاصيلهم ونشاطاتهم والأغراض التي يستعملونها، عبر تحليل تعابير الوجه والصوت، تم إنشاؤها في عام 2009. لها العديد من التطبيقات على نطاق واسع، أيضاً في تطبيقات الرعاية الصحية، والصحة العقلية، علم الروبوتات، وفهم المحادثات، والتعليم، والألعاب، وغيرها الكثير.

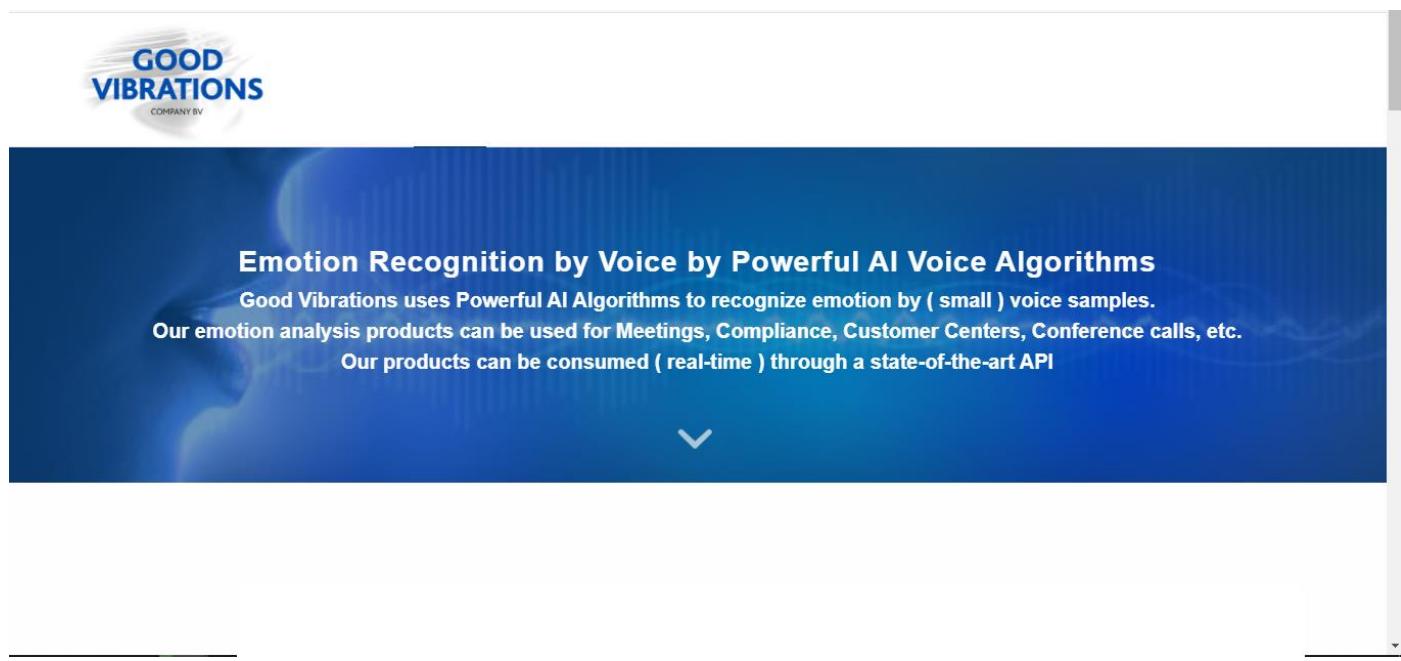
لكن أهم تطبيقاتها هو ما يسمى بال (*Automotive AI*، وقد تم إطلاقه في 21 آذار سنة 2018، وهو أول جهاز استشعار متعدد الأشكال، يوضع داخل كيبلة السيارة لفهم ماذا يحصل للأشخاص في السيارة. ويتم استخدام الكاميرا في السيارة للقياس في الزمن الحقيقي، فعلى سبيل المثال تستطيع هذه التقنية التعرف على حالة السائق، وحالة الركاب، وحالة المركبة بشكل عام. ويمكن أن تسهل الكثير من الأمور للشركة المصنعة للسيارة، وأيضاً من أجل زيادة الأمان في السيارات ودراسة أسباب الحوادث. وأيضاً تستخدم من أجل إنشاء بيئة أكثر راحة وتعطي تجربة ممتعة للتنقل، من خلال فهم كيف يتفاعل الركاب مع البيئة المحيطة بهم، مثل ما يمكنهم فعله في الجزء الخلفي من السيارة، تقديم وجبات خفيفة، تقديم مشروبات، وغيرها الكثير.

بالإضافة لفهم مشاعر السائق والركاب وحالتهم البدنية، فيمكن لهذه التقنية أيضاً أن تكتشف محتوى السيارة، مثل عدد الركاب وأماكنهم، ومكان جلوسهم، وإذا كان هناك شيء غريب موجود في السيارة.

إن شركة (*Affectiva*) تعمل مع العديد من مصانع السيارات الكبيرة وشركات النقل التقنية، مثل:

*Aptiv, Cerence, Hyundai, Kia, Faurecia, Porche, BMW, GreenRoad Technologies and Veoneer*⁶³

2. GOOD – VIBRATIONS:



وهي مختصة بالتعرف على المشاعر من خلال الصوت، عبر خوارزميات الذكاء الاصطناعي للتعرف على الصوت. وقد تم إنشاء هذه الشركة في عام 2011.

هذه الشركة تستخدم خوارزميات ذكاء اصطناعي ذات كفاءة عالية، من أجل التعرف على المشاعر من خلال عينات صوتية صغيرة.

يمكن استخدام أدوات تحليل هذه الشركة من أجل المجتمعات، مراكز خدمة العملاء، ويمكن استخدام تقنيات هذه الشركة في الزمن الحقيقي.

الفصل الثالث

الدراسة التحليلية

أولويات المشروع:

التجربة وقياس الدقة: Performance & accuracy measuring

إن الهدف من المشروع هو تجربة العديد من نماذج وبني الشبكات العصبية، قياس أداء كل منها. وتوثيق النتائج.

الزمن: Time

قدرة المشروع على الإنتهاء قبل موعد التسليم وتقيد المشروع ضمن عامل الوقت كان مفصلاً أساسياً. حيث أن هذا النوع من المشكلات يحتاج إلى الكثير من التجارب والبحث من أجل الحصول على دقة مقبولة نسبياً.

الكلفة: Cost

الكلفة غير مهمة لعدم وجودها، حيث تم تنفيذ المشروع من خلال الأجهزة الخاصة دون الحاجة إلى أي شيء يمكن شراؤه.

	Performance	Time	Cost
Constrain	x	x	
Enhance			
Accept		x	

إدارة المشروع:

تلعب إدارة المشروع دوراً أساسياً في تنظيمه ومراقبة سير العمل من خلال معرفة كل الأنشطة المطلوبة ودرجة تنفيذها والعمل بها والسير ضمن جدول سير العمل لتحقيق النتيجة المطلوبة بأفضل صورة وضمن الوقت المحدد. وقد كان الهدف من التخطيط، تسليم المشروع في الفترة المعلنة للعمل من تاريخ 1/5/2021 وحتى مدة أقصاها 31/3/2022.

منهجية العمل:

منهجية عملنا اعتمدت على البحث، والتعلم من خلال التجربة. ومن أجل مشروعنا قد استخدمنا نموذج الشلال.

(Waterfall Model) في بناء العمليات المتسلسلة، مع إضافة بعض التعديلات بما يناسب نوع العمل اللازم لهذا المشروع. حيث كانت القدرة في التحكم بالمشروع من خلال سير العملية التطويرية للمشروع والانتهاء من كل عملية على حدة للانتقال للعملية التالية وحتى انتهاء المرحلة، ومنه للمرحلة التالية. لم تشهد مرحلة تطوير المشروع أي تغيير أو تعديل سابق أو لاحق لأوانه مما دل على القدرة العالمية في التحكم بالمشروع خطوات والإدارة الصحيحة من قبل

أعضاء الفريق، وهذا كان له انعكاس إيجابي على تقديم محتوى ذات جودة عالية دون أخطاء تذكر. نظراً لأن متطلبات المشروع واضحة وثابتة لم تتغير منذ بداية المشروع، والتقنية المستخدمة في المشروع مفهومة إلى حد ما، بالإضافة لكونها محددة منذ البداية كما أنه لا يوجد متطلبات غامضة حيث أننا نقوم بدراسة كل حالة ونقوم بتسجيل النتائج التي أحرزتها.

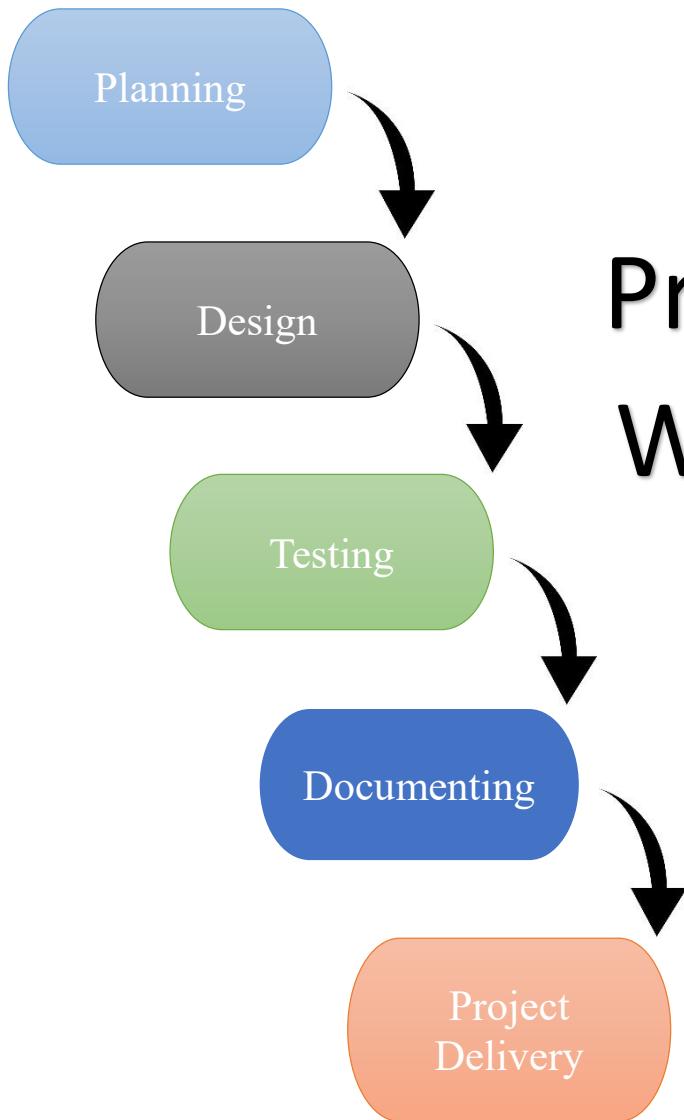
أسباب استخدام نموذج الشلال:

- المتطلبات واضحة ومفهومة.
- المتطلبات موثقة بشكل جيد وثابتة.
- التقنيات مفهومة إلى حد ما.
- زمن تنفيذ المشروع متوسط نسبياً.

مميزات نموذج الشلال:

- تسلسل الإجرائيات خطى.
- درجة التحكم في إجرائيات البحث العلمي والتجربة عالية بحكم وجود المراحل ذات المهام المحددة والمعرفة مسبقاً.
- درجة التوثيق عالية.
- تقييم كل نموذج في نهاية كل عملية تنفيذ.
- تسلسل الإجرائيات باتجاه وحيد.

لقد قمنا بضم عملية التصميم والبحث والتجربة في مرحلة واحدة، فضلاً عن تحديد المتطلبات في عملية التخطيط، بالإضافة إلى عملية الاختبار والتوثيق، وانتهاءً بتسليم المشروع.



Project based on Waterfall Model

هيكل تجزئة المشروع:

تعتبر من الأدوات التي تمكن مدير المشروع من تقسيم العمل الى وحدات صغيرة تجعل القدرة على التحكم بالمشروع أكثر سهولة من الطرق التقليدية. حيث ان **Work Breakdown Structure** يحتوي على المراحل التالية:
 المستوى الأول: هو اسم المشروع.

المستوى الثاني: يحوي على أجزاء دورة حياة المشروع الموضحة سابقاً.

كل مرحلة تحوي على مجموعة من العمليات الصغيرة التي يتم العمل بها للانتقال الى المرحلة التالية من المشروع.

تم وضع نقاط علام (♦) قبل الانتهاء من كل مرحلة للدلالة من التحقق من المخرجات السابقة والتأكد من المحتوى الخاص من كل عملية يراجع جميع أعضاء الفريق المخرجات وتم الموافقة عليه للانتقال للمرحلة التالية. في حال وجد

اعتراض من قبل أحد أعضاء الفريق على المحتوى، تتم المناقشة بين أعضاء الفريق للوصول الى حل بديل يكون أفضل وحاصل على موافقة الجميع.

أولاً: مرحلة التخطيط:

1. التعرف على مفاهيم المشكلة.
2. كتابة الدراسة الأولية.
3. التعرف على أنواع الشبكات العصبية.
4. التعرف على فواعد البيانات (*datasets*) التي تتعلق بالمشكلة المطروحة في مشروعنا.
5. التعرف على خوارزميات وتطبيقات سابقة تتعلق بفكرة مشروعنا.
6. انتهاء مرحلة التخطيط. (نقطة علام) (♦).

ثانياً: مرحلة التصميم:

1. بناء نماذج شبكة عصبية التفافية للتعرف على صنفين من المشاعر (الحزن والسعادة).
2. بناء نماذج شبكة عصبية التفافية من أجل المشاعر السبعة (الغضب – الاشمئاز – الخوف – السعادة – الحزن – الحياديّة – المفاجأة).
3. بناء نماذج تعلم عميق باستخدام ال (*Transfer Learning*).
4. بناء نماذج تعلم عميق باستخدام شبكة (*EfficientNet*).
4. تجربة ال (*Feature Extraction*) في نماذج التعلم العميق.
5. تجربة ال (*Fine – Tuning*) في نماذج التعلم العميق.
6. تجربة ال (*scaling – up*) في نماذج التعلم العميق.
7. التعرف على شبكة (*MobileNet*)، وأنواعها ومزاياها.
8. بناء النموذج الأخير.
9. انتهاء مرحلة التصميم. (نقطة علام) (♦).

ثالثاً مرحلة الاختبار:

1. تقييم نماذج الشبكات العصبية التفافية المبنية لتصنيف نوعين من المشاعر.
2. تقييم نماذج الشبكات العصبية التفافية المبنية لتصنيف المشاعر
3. تقييم نماذج الشبكات العصبية المبنية على ال (*Transfer learning*).
4. الانتهاء من مرحلة الاختبار. (نقطة علام) (♦).

رابعاً: مرحلة التوثيق:

1. كتابة دراسة باللغة العربية، وهي عبارة عن مقارنة لخوارزميات سابقة لتنفيذ مشروعنا.
2. كتابة تقرير عن النتائج التي حصلنا عليها خلال التنفيذ.
3. كتابة تقرير المشروع.
4. عمل فيديو تعريفي عن المشروع.
5. جمع الملفات. (نقطة علام) (♦).

الدراسة التخطيطية:

تعتبر عملية التخطيط من اهم الاشياء التي يجب مراعاتها اثناء القيام بأي مشروع صغيراً كان ام كبيراً، وانطلاقاً من هذه الفكرة اولينا عملية التنظيم جزءاً جيداً من وقت المشروع، لنتمكن من وضع خطة محدمة التنفيذ نستطيع من خلالها تفادي الوقوع في الأخطاء او تداخل العمل مما قد يؤدي الى تكبد المشروع خسائر تكون صورتها على شكل وقت لا يمكن تعويضه.

وضعت خطة أولية من أجل منهجية عمل المشروع كالتالي:

- 1.أخذ عينات من مجموعات البيانات بشكل عشوائي.
- 2.بناء الشبكة العصبية وتدريبها.
- 3.تسجيل النتائج ومقارنتها.
- 4.محاولة تحسين النتائج التي حصلنا عليها.

موارد المشروع:

الموارد البشرية:

يتكون فريق العمل من الطلاب:

1. محمد تميم عليوي
2. تاج الدين الموصلي

بإشراف:

الدكتورة أميمة دكاك.

المهندسة منار منعم.

الموارد الفизيائية:

حاسوبان، واتصال بالإنترنت.

بيئة العمل:

1. *Anaconda*.

2. *Jupyter notebook*

3. *GoogleColab*

لغات البرمجة والمكاتب:

1. *Python 3.X*.

2. *TensorFlow*.

3. *keras*.

4. *OpenCV*.

5. *Scikit – Learn*

ادارة المخاطر:

تعتبر إدارة المخاطر من الأشياء المهمة التي يجب مراعاة الأخذ بها تفاديًّا لحدوث الأخطاء وتجنب إضاعة الوقت والتأثير على جودة المشروع. وهنا يجب ذكر ما تمت مراعاته والأخذ بالحسبان منه والرصد الجيد للأخطاء وتجنب الأثر الكارثي على سير العمل.

تم وضع هذا الجدول من أجل الذي يوضح المخاطر ومدى تأثيرها على خطة العمل، بالإضافة الى الاجرائيات المتبعة لتفادي الوقوع بها، وحرصاً على استخدام الحل الأمثل في معالجة الأخطار.

نوع المخاطرة	المخاطرة	احتمال الحدوث	نسبة التأثير	الأثر على أهداف المشروع	طرق التفادي
مخاطرة خارجية	انقطاع التيار الكهربائي	كبير	كبيرة	تأخير في تسليم المهام الجزئية وانعكاسه على تسليم المشروع.	شراء بطاريات جديدة بالإضافة إلى وحدات شحن خارجية.
مخاطرة خارجية	انقطاع/ ضعف الانترنت	متوسط	كبيرة	ضعف التواصل بين أعضاء الفريق، عدم التمكن من الدخول إلى بيئة تدريب النماذج	الذهاب إلى مقاصف أو مقاهي انترنت أو تفعيل باقات انترنت.
مخاطرة خارجية	عطل في الأجهزة	متوسط	كبيرة	تأخير في تسليم المهام الجزئية نظراً لوجود عطل في أحد أجهزة أعضاء الفريق.	العمل من خلال أجهزة قديمة تفي بالغرض لفترة محددة.
مخاطرة خارجية	انشغال في عمل اخر	كبير	كبيرة	ضيق الوقت في العمل على المشروع.	العمل في فترات نهاية الأسبوع والشهر المتواصل.
مخاطرة خارجية	تسليم وظائف المواد الجامعية	كبير	كبيرة	انشغال في تسليم وظائف المواد وضيق الوقت من أجل العمل على المشروع.	العمل في فترات نهاية الأسبوع والشهر المتواصل.
مخاطرة تنظيمية	بعد فريق العمل	كبير	متوسطة	ضعف التواصل بين أعضاء الفريق وطول المدة الزمنية في اتخاذ القرار.	اجراء جلسات افتراضية بين أعضاء الفريق بالإضافة إلى مجموعة حوار للنقاش واتخاذ القرار.
مخاطرة تنظيمية	التقدير الغير دقيق لبعض اجراء عمليات المشروع	متوسط	كبيرة	التأخير في تسليم بعض المراحل الجزئية للمشروع.	تجزئة المهام الجزئية إلى مهام جزئية افتراضية أصغر منها للمتابعة الأفضل لسير العمل في المهمة الجزئية.

التأكد من مدى كفاءة قواعد سواءً يدوياً أو من خلال برمجيات مخصصة لهذا الغرض	أن تكون المعطيات معطوبة جزئياً، وهذا يؤدي إلى نتائج غير صحيحة	كبيرة	متوسط	فساد الصور في قواعد المعطيات (Datasets)	مخاطر فنية
	عدم استطاعة المستخدم العادي من استخدام نموذج التعلم العميق لتجربته على صوره الخاصة.	كبيرة	كبيرة	عدم القدرة على جعل النموذج قابل للاستخدام من قبل المستخدم	مخاطر فنية

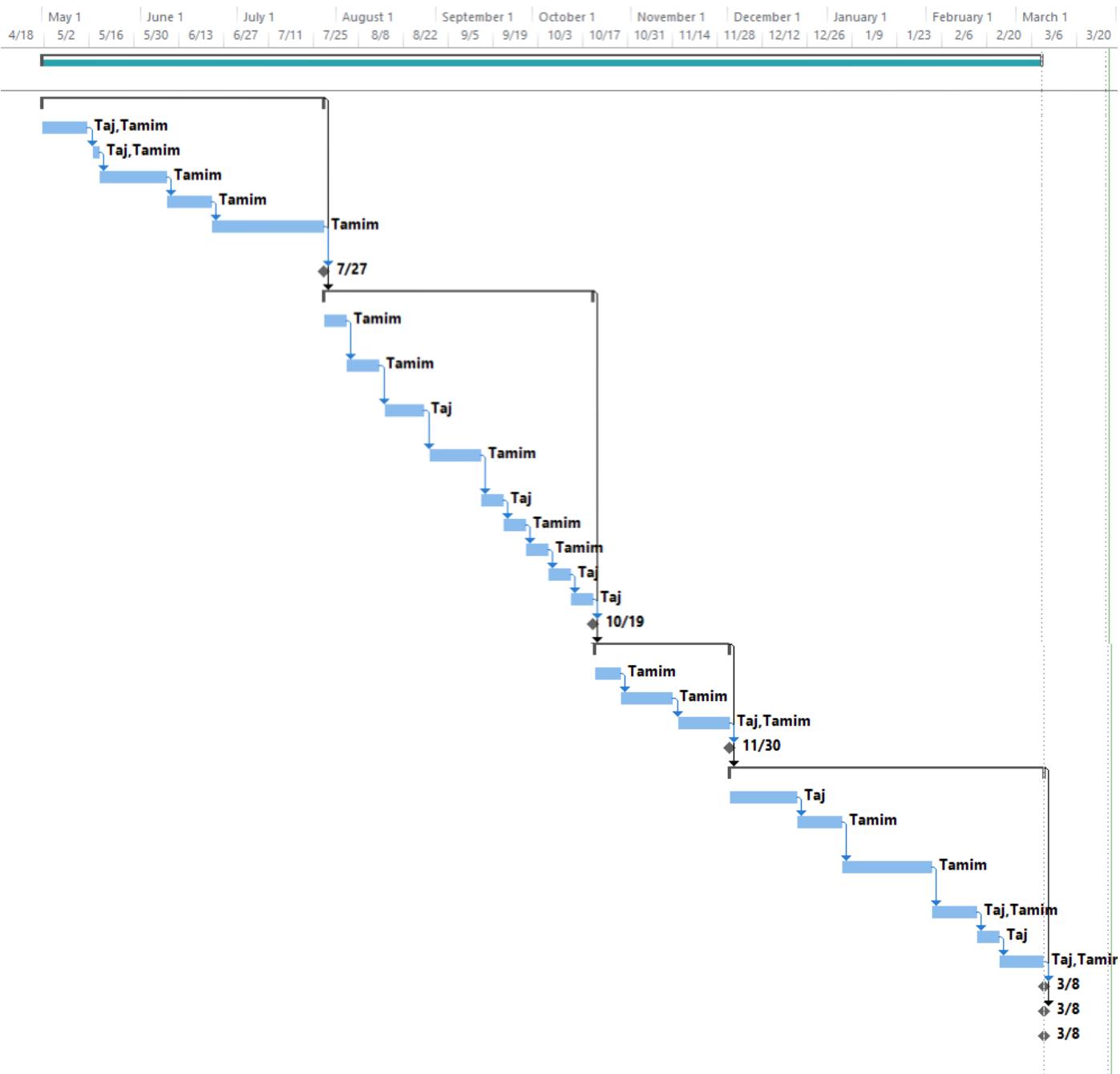
مخطط الأنشطة الإجرائية:

استخدمنا في عمل مراحل المشروع برنامج MS Project الذي يمكن من خلاله استخدام مخطط غantt في تقسيم المهام الفرعية للمشروع وتحديد من القائم عليها والمدة الكلية لتنفيذ المشروع بالإضافة الى الفترات الزمنية لكل عملية جزئية. وفي حال كانت هناك كلف او رواتب تصرف يمكن اضافتها بسهولة في الموارد على أنها أجراً ساعي للموظف او كمورد ثابت مثل الحاسب او المخدم. سهلت هذه الآلة الكثير من الوقت على الفريق بالإضافة الى كونها أحد اهم الإجرائيات المتتبعة في المشروع وذات نفاذ قوي في الوقت. تعمل كمرجع دائم من حيث المدة المتبقية على الاجرائية للانتهاء والوقت الكامل لجزء من المشروع.

وتم العمل على المشروع حيث انه:

- لا يوجد أيام عطل في نهاية الأسبوع.
- لا يوجد عدد ساعات محددة للعمل (كل ساعة فارغة، هي ساعة عمل).
- لا يوجد عطل رسمي.
- يبدأ المشروع في 1/05/2021 وينتهي في 8/03/2022.
- يحتوي اسم القائم على العمل خلال المهمة الجزئية.

WBS	Task Name	Duration	Start	Finish	Resource Names
1	« Deep learning for facial emotion recognition	223 days	Sat 5/1/21	Tue 3/8/22	Taj, Tamim
1.1	« Planning	63 days	Sat 5/1/21	Tue 7/27/21	Taj, Tamim
1.1.1	Exploring the problem concepts	11 days	Sat 5/1/21	Fri 5/14/21	Taj, Tamim
1.1.2	Writing the Initail Study	2 days	Mon 5/17/21	Tue 5/18/21	Taj, Tamim
1.1.3	Exploring the kinds of nural netw	15 days	Wed 5/19/21	Tue 6/8/21	Tamim
1.1.4	Exploring available datasets	10 days	Wed 6/9/21	Tue 6/22/21	Tamim
1.1.5	Exploring previous algorithms & applications	25 days	Wed 6/23/21	Tue 7/27/21	Tamim
1.1.6	Finish Planning	0 days	Tue 7/27/21	Tue 7/27/21	
1.2	« Design	60 days	Wed 7/28/21	Tue 10/19/21	Taj, Tamim
1.2.1	Buildings CNNs models for two emotions	5 days	Wed 7/28/21	Tue 8/3/21	Tamim
1.2.2	Building CNNs for the seven mentioned emotions	8 days	Wed 8/4/21	Fri 8/13/21	Tamim
1.2.3	Building models with transfer learning	10 days	Mon 8/16/21	Fri 8/27/21	Taj
1.2.4	Building models with EfficientNet networks	12 days	Mon 8/30/21	Tue 9/14/21	Tamim
1.2.5	Trying feature extraction	5 days	Wed 9/15/21	Tue 9/21/21	Taj
1.2.6	Trying Fine-tunning	5 days	Wed 9/22/21	Tue 9/28/21	Tamim
1.2.7	Trying scalling-up	5 days	Wed 9/29/21	Tue 10/5/21	Tamim
1.2.8	Exploring MobilNet networks	5 days	Wed 10/6/21	Tue 10/12/21	Taj
1.2.9	Building Final model	5 days	Wed 10/13/21	Tue 10/19/21	Taj
1.2.10	Finish Designing	0 days	Tue 10/19/21	Tue 10/19/21	
1.3	« Testing	30 days	Wed 10/20/21	Tue 11/30/21	
1.3.1	Evaluate CNNs Binary models	6 days	Wed 10/20/21	Wed 10/27/21	Tamim
1.3.2	Evaluate CNNs Models	12 days	Thu 10/28/21	Fri 11/12/21	Tamim
1.3.3	Evaluate EfficientNet Models	12 days	Mon 11/15/21	Tue 11/30/21	Taj, Tamim
1.3.4	Finishing Testing	0 days	Tue 11/30/21	Tue 11/30/21	
1.4	« Documenting	70 days	Wed 12/1/21	Tue 3/8/22	Taj, Tamim
1.4.1	Writing a comparison paper abou	15 days	Wed 12/1/21	Tue 12/21/21	Taj
1.4.2	Writing a comparison paper about the Results	10 days	Wed 12/22/21	Tue 1/4/22	Tamim
1.4.3	Writing a Study about previous algorithms	20 days	Wed 1/5/22	Tue 2/1/22	Tamim
1.4.4	Writing the Project report	10 days	Wed 2/2/22	Tue 2/15/22	Taj, Tamim
1.4.5	Making a PowerPoint Presentatic	5 days	Wed 2/16/22	Tue 2/22/22	Taj
1.4.6	Making an Explinationary video	10 days	Wed 2/23/22	Tue 3/8/22	Taj, Tamim
1.4.7	Collecting project Folders	0 days	Tue 3/8/22	Tue 3/8/22	
1.5	« Project Submitting	0 days	Tue 3/8/22	Tue 3/8/22	
1.5.1	Upload Project folders to LMS	0 days	Tue 3/8/22	Tue 3/8/22	



مصفوفة المسؤوليات : (RM) Responsibility Matrix

تستخدم مصفوفة المسؤوليات في توضيح دور كل عضو من أعضاء الفريق في المشروع، ويبين مساهمنته في المهمة والدور الذي كان يعمل عليها، وتوضح المصفوفة مساهمة كل عضو على شكل:

- ♦ **مسؤول (R):** هو الشخص الموكلة له هذه المهمة ويقع على عاتقه تسليمها.
- ♦ **مساعد (S):** هو الشخص الذي يبقى على اطلاع على سيرورة العملية والمسؤول عن إعطاء الحلول البديلة في حال استعصى على المسؤول ايجاده.
- ♦ **بيان الجدول التالي المهام الجزئية الموكلة لكل عضو من أعضاء الفريق:**

Students		Activity	WBS
Taj	Tamim		
الخطيط			1.1
R	R	التعرف على مفاهيم المشكلة	1.1.1
R	R	كتابة الدراسة الأولية	1.1.2
R	R	التعرف على أنواع الشبكات العصبية	1.1.3
S	R	التعرف على قواعد البيانات (datasets) الموجة	1.1.4
R	R	التعرف على خوارزميات وتطبيقات سابقة تتعلق بفكرة المشروع	1.1.5
		انهاء الخطيط	1.1.8
التصميم			1.2
تصميم نماذج للشبكات العصبية بأشكالها			1.2.1
S	R	بناء نماذج تعلم شبكة عصبية تقافية للتعرف على صنفين من المشاعر (حزن وغضب)	1.2.1.1
S	R	بناء نماذج تعلم شبكة عصبية تقافية للتعرف على المشاعر السبعة.	1.2.1.2
S	R	بناء نماذج تعلم عميق باستخدام ال (Transfer Learning)	1.2.1.3
S	R	بناء نماذج تعلم عميق باستخدام شبكتي (EfficientNet – Resnet50)	1.2.1.3.1
S	R	تجربة استخراج السمات	1.2.1.3.2
S	R	تجربة الضبط الدقيق	1.2.1.3.3
R	S	بناء النموذج الآخر	1.2.2
		انتهاء التصميم	1.2.3
الاختبار			1.3
S	R	تقييم نماذج الشبكة العصبية الالتفافية الثانية	1.3.1
S	R	تقييم نماذج الشبكة العصبية الالتفافية للمشاعر السبعة	1.3.2
S	R	تقييم نماذج استخراج السمات والضبط الدقيق (Transfer Learning) باستخدام ال	1.3.3
R	R	تقييم النموذج الآخر	1.3.4
		انتهاء الاختبار	1.3.5
الوثيق			1.4
S	R	كتابة دراسة باللغة العربية، وهي عبارة عن مقارنة لخوارزميات سابقة لتنفيذ مشروعنا	1.4.1
R	R	كتابة تقرير عن النتائج التي حصلنا عليها	1.4.2
S	R	كتابة تقرير المشروع	1.4.3
S	R	عمل فيديو التعريف	1.4.4
S	R	جمع الملفات	1.4.5
تسليم المشروع			1.5
		رفع المحتوى على نظام التعليم	1.5.1

الفصل الرابع

التقانات 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. بشكل عام، يمكن استخدام بايثون لعمل البرامج البسيطة للمبتدئين، وإنجاز المشاريع الضخمة في الوقت نفسه. غالباً ما ينصح المبتدئون في ميدان البرمجة بتعلم هذه اللغة لأنها من بين أسرع اللغات البرمجية تعلمًا.

تتميز بايثون بمجتمعها النشط، كما أن لها الكثير من المكتبات البرمجية ذات الأغراض الخاصة التي برمجها أشخاص من ذلك المجتمع. مثلاً، هناك مكتبة Pygame التي توفر مجموعة من الدوال من أجل برمجة الألعاب. يمكن لبايثون أيضاً التعامل مع العديد من أنواع قواعد البيانات مثل MySQL وغير ذلك.

تدعم بايثون أنماط برمجية متعددة هي التوجيه الكائني، البرمجة 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 Cpython.



2. *Anaconda*

هي عبارة عن توزيعه بايثون خاصة تأتي مجانية أو مدفوعة الأجر هدفها الرئيسي هو تسهيل إدارة مكتبات معينة ومحددة للغة بايثون. المكتبات التي تتحدث عنها في بايثون اناكوندا هي مكتبات مختصة بمعالجة البيانات الضخمة، وتطبيقات الحوسبة العلمية وغيرها من المكتبات العلمية. تعتبر توزيعة *Anaconda* من توزيعات لغة *python* المشهورة في مجال علوم البيانات، حيث تجد كثيراً من المقالات العلمية تتحدث عنها عند استخدام لغة *python* في معالجة البيانات وتعلم الآلة والتعلم العميق ومعالجة اللغات الطبيعية.



3. Jupyter notebook

تُعد أداة **Jupyter Notebook** أحد أشهر وأهم الأدوات المستخدمة أثناء تحليل البيانات. بالإضافة إلى ذلك، هي شيء أساسي لا يستغني عنها عالم البيانات، وذلك لما تقدمه من مميزات وخصائص تسهل من التعامل مع البيانات والشيفرة البرمجية. تُعتبر **Jupyter** أداة قوية يمكن استخدامها تفاعليًا في مشاريع علم البيانات وتعلم الآلة والتعلم العميق.

Jupyter notebook هو تطبيق ويب يتيح للمستخدمين الجمع بين النص التوضيحي والمعادلات الرياضية والتعليمات البرمجية والمرئيات في مستند واحد سهل المشاركة.

أصبح **Jupyter** بسرعة أداة لا غنى عنها لتحليل البيانات والتعلم الآلي والعاملين الآخرين في تنظيف البيانات واستكشافها. يمكن أن يجمع جميع المواد ذات الصلة مثل المستندات والرمز في مكان واحد، حتى يتمكن المستخدمون من الرؤية في لمحه.

Jupyter مشتق من **IPython** وهو غلاف تفاعلي مشابه لغلاف **Python** عادي. في الأصل، عملت دفاتر الملاحظات عن طريق إرسال رسائل من تطبيق ويب (دفتر الملاحظات الذي رأيته في المستعرض الخاص بك) إلى **IPython kernel** (تطبيق **IPython** يعمل في الخلفية). تقوم النواة بتنفيذ التعليمات البرمجية وإرسال النتائج مرة أخرى إلى دفتر الملاحظات. بعد التطور التقني لـ**IPython**، يتم فصل النواة إلى النموذج المعماري الحالي: جوهر العمارة هو خادم دفتر الملاحظات. يتصل المستخدم بالخادم من خلال متصفح. بعد إرسال الرمز المكتوب من قبل المستخدم في تطبيق الويب إلى خادم دفتر الملاحظات من خلال المتصفح، يرسل خادم دفتر الملاحظات الرمز إلى النواة، وتعمل النواة وترسل النتيجة إلى خادم دفتر الملاحظات. وأخيراً، يتم إرجاع الإخراج إلى المستعرض وعرضه للمستخدم.

عند حفظ دفتر الملاحظات، يتم إنشاء كل شيء في ملف JSON (بامتداد الملف.ipynb). وكتابته إلى خادم دفتر الملاحظات.



4. Google Colab

Colaboratory هو مشروع بحثي من Google مصمم للمساعدة في نشر تدريب التعلم الآلي ونتائج البحث. إنها بيئة دفتر ملاحظات Jupyter يمكن استخدامها بدون أي إعدادات، و تعمل بالكامل في السحابة (Google Cloud).

يتم تخزين الدفاتر (Notebooks) في Google Drive ويمكن مشاركتها تماماً مثلما تستخدم مستندات Google أو جداول البيانات. Colaboratory مجاني للاستخدام.

باستخدام Colaboratory، يمكنك بسهولة استخدام Keras و TensorFlow و PyTorch و OpenCV وأطر أخرى لتطوير تطبيقات التعلم العميق.

بالمقارنة مع الخدمات السحابية الأخرى، فإن الميزة الأكثر أهمية هي أنه يقوم بتوفير GPU Colab وهو مجاني تماماً.



إن وحدة معالجة الرسوميات التي تستخدم في (Colab) تدعى ب (NVIDIA Tesla K80).



TensorFlow

5. *TensorFlow*:

يُعتبر "TensorFlow" الجيل الثاني من نظام تعلم الآلة لـ *Google*, فهو مكتبة برمجيات مفتوحة المصدر للتعلم الآلي، حيث يستخدم في العديد من المنتجات التجارية لشركة جوجل؛ لتميزه بالمرونة، وسهولة الاستخدام والتطبيق، والسرعة، بالإضافة إلى أهم ما يميزه وهو كونه مفتوح المصدر للجميع.

لذلك "TensorFlow" لديه قدرة عالية جدًا على البحث والرد الذكي، وتستخدمه جوجل في خدماتها المختلفة، بل الان أصبح يستخدم في منتجات حقيقة؛ وذلك لأنّ النظام يبني على أساس الاستعداد الكامل لخدمة الإنتاج، لسرعته والقدرة على استخدامه، فقد بني على مصدر مفتوح يتيح وصول أي شخص للرموز البرمجية المصدرية وتعديلها، حيث يقوم النظام بتمرير البيانات المعقدة من خلال الشبكة العصبية، والتي تعتبر جزءاً أساسياً في التعليم العميق.

"ولكن أهم ما يميز "TensorFlow" بالنسبة للمبرمج هو سهولة التعبير عن أفكاره من خلاله باستخدام *Python* لغة"

"A general purpose, high – level language" يتم تعريفها تحت مسمى "python" وبما أنّ لغة "python" لأنّها لغة البرمجة التي تشمل برمجة كل شيء مثل: الألعاب، المواقع، قواعد البيانات، برمجة الآلات، وأيضاً برمجة تطبيقات الموبايل. بالإضافة إلى أنها "An interpreted language"

لأن أي لغة برمجة "high – level" ، تقم بكتابتها الحاسوب لا يفهمها مباشرةً، لذلك لابد أن يتم تحويلها إلى ما يسمى بلغة الآلة "machine language" حتى يفهمها الحاسوب ويقوم بالتعامل معها، ولغة الآلة عبارة عن "1/0" وبالتالي فانت تحتاج لوسيط يقوم بتحويل لغة البرمجة التي قمت بكتابتها لـ لغة الآلة، في حالة لغات برمجة مثل (C, C++) هذا الوسيط يسمى "compiler" ، وهو ما يحتاج إلى برامج أخرى تقم بتنزيلها من أجل تشغيل الرمaz البرمجي.

ولكن في حالة لغة "Python" يسمى هذا الوسيط "Interpreter" ، والذي يعطيك ميزة التسهيل فكل ما تحتاج إليه تنزيل ما يعرف باسم "IDE" ، لكتابة الرمaz البرمجي وضغط زر "run" من أجل التشغيل، وهذه النقطة تُعطِي لغة (Python) ميزة قوية جدًا وهي أنها "cross platform" ، حيث أنها تقبل تشغيل الرمaz البرمجي.



6. OpenCV:

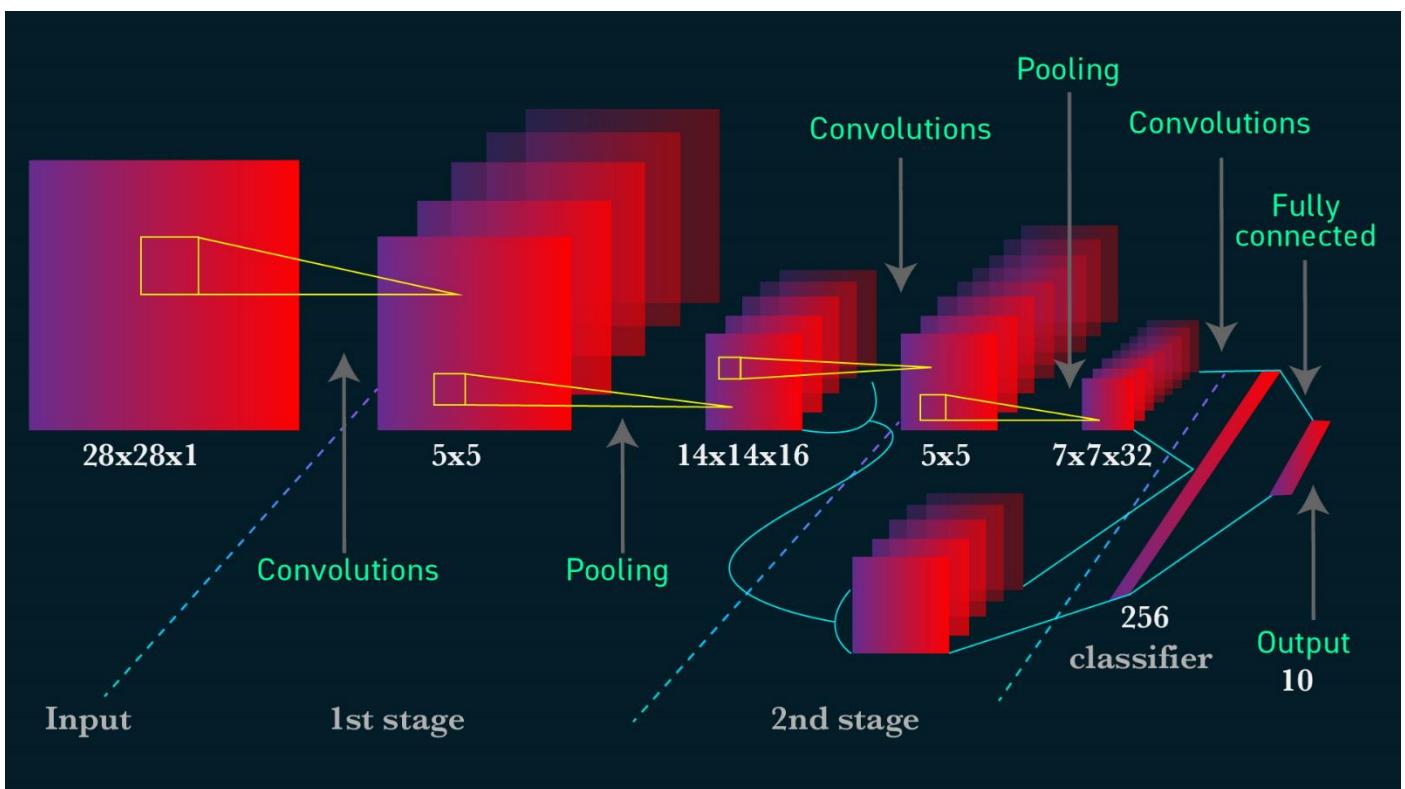
(OpenCV). أو المكتبة البرمجية المفتوحة للرؤية الحاسوبية هي مكتبة برمجية تهدف بشكل أساسى لتطوير الرؤية الحاسوبية، طورتها شركة أنتل (Intel) وتعاقب على دعمها بعد ذلك شركة (Willow Garage) للروبوتات و (Itseez) وأخير شركة (Intel) مجدداً بعد استيلانها على شركة (Itseez) منذ 2016. المكتبة مجانية كونها تحت رخصة المصدر المفتوح (open source BSD license). ويمكن استخدامها على معظم الأنظمة الحاسوبية التي تدعم لغتي (C, C++). تركز بشكل أساسى على معالجة اللحظية للصور (real – time).

إذا وجدت المكتبة على النظام ما يدعى بشكل الأداء المتكامل (Integrated Performance Primitives) المطورة من قبل شركة أنتل (Intel) فسوف تستخدم هذه الإجراءات التحسينية ذات الرخصة الخاصة للتسرير من عملها.

تحتوي مكتبة (OpenCV) على عدة أدوات وهي:

وحدة نمطية مدمجة لتحديد هيكل البيانات الأساسية، المستخدمة من قبل جميع وحدات أخرى. .

- ***imgproc***: وحدة معالجة الصور التي تتضمن صورة الخطية وغير الخطية والترشيح، والتحولات صورة هندسية (تغيير الحجم، تزييفها أفيني والمنظور، عام الخارطة المستندة إلى الجدول)، وتحويل لون الفضاء، رسوم بيانية، وخصائص أخرى.
- ***video***: وحدة تحليل الفيديو التي تتضمن تقدير الحركة، خلفية الطرح، خوارزميات تتبع الأغراض.
- ***calib3d***: متعددة مشاهدة الخوارزميات الهندسية الأساسية، واحد وستيريو معايرة الكاميرا، وجوه تشكيل تقدير، خوارزميات المراسلات ستيريو، وعناصر إعادة الإعمار D.3
- ***objdetect***: كشف عن وجوه وحالات من الفئات المحددة مسبقاً (على سبيل المثال، وجوه، عيون، وأكواب، والناس، والسيارات).
- ***highgui***: وهي سهلة الاستخدام واجهة لانتقاد الفيديو والصورة والفيديو الترميز، فضلاً عن قدرات واجهة المستخدم بسيطة.



7. Convolutional Neural Network:

في تعلم الآلة، يقوم المصنف بتخفيص اسم صف لمجموعة بيانات صور، على سبيل المثال: ينتج مصنف صور أسماء واصفة لمجموعة بيانات صور، مثلًا (طائرة، طير، ...الخ)

الشبكة العصبية التلقيفية: خوارزمية تستخدم للتعرف على الأنماط في البيانات. تتكون الشبكات العصبية بشكل عام من مجموعة من الخلايا العصبية التي يتم تنظيمها في طبقات، وكل منها أووزانها وتحيزاتها القابلة للتعلم. من أجزائها الأساسية:

Tensor – Neuron – Layer – Kernel weights and biases

طبقات الشبكة العصبية التلفيفية:

طبقة الدخل: لنفترض استخدامنا لصور (RGB)، فإن طبقة الإدخال لدينا بها ثلاثة قنوات، تتوافق مع قنوات الألوان: الأحمر والأزرق والأخضر، على التوالي.

الطبقات التلفيفية: هي أساس هذا النوع من الشبكات العصبية، لأنها تحوي على النواة المكتسبة (*weights*)، والتي تستخرج السمات التي تميز الصور المختلفة عن بعضها البعض، وهذا ما نريده للتصنيف. يقوم العصبون التلفيفي بتنفيذ منتج نقطي عنصري بنواة فريدة وإخراج الخلايا العصبية المقابلة للطبقة السابقة. سيؤدي هذا إلى العديد من النتائج الوسيطة حيث توجد (kernels) فريدة من نوعها. العصبون التلفيفي هو نتيجة كل النتائج الوسيطة التي تم جمعها مع التحيز المكتسب.

معاملات الشبكة العصبية التلفيفية:

1. Padding:

غالباً ما يكون ال (*Padding*) ضرورياً عندما يمتد ال (*kernel*) إلى ما بعد خريطة التنشيط (*activation map*). يحافظ ال (*Padding*) على البيانات عند حدود خرائط التنشيط، مما يؤدي إلى أداء أفضل، ويمكن أن يساعد في الحفاظ على الحجم المكاني (*spatial size*) للإدخال، مما يسمح لمصمم الهندسة المعمارية ببناء شبكات ذات أداء أعلى. توجد العديد من تقنيات الحشو، ولكن الطريقة الأكثر شيوعاً هي (*approached zero – padding*) نظراً لأنها وبساطتها وكفاءتها الحسابية. تتضمن التقنية إضافة أصفار بشكل متماثل حول حواف الإدخال. تم اعتماد هذا النهج من قبل العديد من شبكات CNN عالية الأداء مثل AlexNet.

2. Kernel Size

حجم النواة غالباً ما يشار إليه أيضاً باسم حجم المرشح، يشير إلى أبعاد النافذة المنزلقة فوق المدخلات. اختيار هذا المعامل الفائق له تأثير كبير على مهمة تصنيف الصورة. على سبيل المثال، يمكن لأحجام النواة الصغيرة استخراج قدر أكبر بكثير من المعلومات التي تحتوي على ميزات محلية عالية من الإدخال. يؤدي حجم النواة الأصغر أيضاً إلى تقليل أبعاد الطبقة، مما يسمح ببنية أعمق. على العكس من ذلك، فإن حجم النواة الكبير يستخرج معلومات أقل، مما يؤدي إلى تقليل أسرع في أبعاد الطبقة، مما يؤدي غالباً إلى أداء أسوأ. تعد النواة الكبيرة أكثر ملائمة لاستخراج الميزات الأكبر حجماً. في النهاية، سيعتمد اختيار حجم النواة المناسب على مهمتك ومجموعة البيانات الخاصة بك، ولكن بشكل عام، تؤدي أحجام النواة الأصغر إلى أداء أفضل لمهمة تصنيف الصور لأن تصميم بنية الشبكة قادر على تكديس المزيد والمزيد من الطبقات معاً تعلم المزيد والمزيد من الميزات المعقدة.

3. Stride

يشير (*Stride*) إلى عدد وحدات البيكسل التي يجب إزاحة النواة عليها في المرة الواحدة. على سبيل المثال، تستخدم Tiny VGG خطوة 1 لطبقاتها التلفيفية، مما يعني أن المنتج النقطي يتم تنفيذه على نافذة (3×3) من المدخلات لإعطاء قيمة خرج، ثم يتم تحويله إلى بمقدار بيكسل واحد لكل عملية لاحقة. خطوة التأثير على (CNN) تشبه حجم النواة. مع انخفاض الخطوة، يتم التعرف على المزيد من الميزات لأنه يتم استخراج المزيد من البيانات، مما يؤدي أيضاً إلى طبقات إخراج أكبر. على العكس من ذلك، مع زيادة الخطوة، يؤدي ذلك إلى استخراج ميزات أكثر محدودية وأبعد طبقة إخراج أصغر. تتمثل إحدى مسؤوليات مصمم بنية الشبكة في التأكد من أن النواة تنزلق عبر المدخلات بشكل متماثل عند تنفيذ ال (CNN).

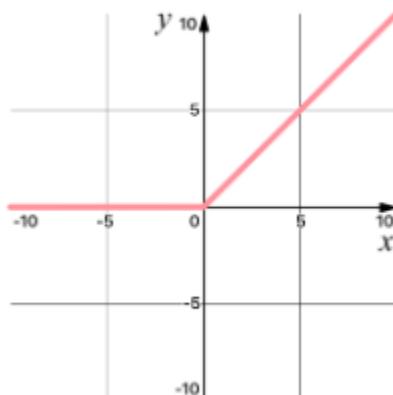
توابع التنشيط (Activation Functions)

1. *ReLU*:

الشبكات العصبية منتشرة للغاية في التكنولوجيا الحديثة - لأنها دقيقة للغاية! تتكون أعلى شبكات (*CNN*) أداءً اليوم من كمية كبيرة جدًا من الطبقات القادرة على تعلم المزيد والمزيد من الميزات. يرجع جزء من سبب قدرة شبكات (*CNN*) الرائدة هذه على تحقيق مثل هذه الدقة الهائلة إلى عدم خطيتها. تطبق *ReLU* اللاخطية التي تمس الحاجة إليها في النموذج. اللاخطية ضرورية لانتاج حدود قرار غير خطية، بحيث لا يمكن كتابة المخرجات كمجموع خطية من المدخلات. إذا لم تكن وظيفة التنشيط غير الخطية موجودة، فإن بنية ال*CNN* العميق ستنتقل إلى طبقة تأفييفية واحدة مكافئة، والتي لن تؤدي بشكل جيد تقريبًا. تُستخدم وظيفة التنشيط *ReLU* على وجه التحديد كوظيفة تنشيط غير خطية، على عكس الوظائف غير الخطية الأخرى مثل *Sigmoid* لأنه قد لوحظ بشكل تجريبي أن *CNNs* التي تستخدم *ReLU* أسرع في التدريب من نظيراتها.

تابع ال (*ReLU*):

$$\text{ReLU}(x) = \max(0, x)$$



يتم تطبيق وظيفة التنشيط هذه على كل قيمة من موتر الإدخال. على سبيل المثال، إذا تم تطبيق *ReLU* على القيمة 2.24، فستكون النتيجة 2.24، نظرًا لأن 2.24 أكبر من 0. يتم تنفيذ وظيفة التنشيط الخطى المصحح (*ReLU*) بعد كل طبقة تأفييفية في بنية الشبكة.

2. *Softmax*:

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

تخدم عملية *softmax* غرضًا رئيسيًا: التأكد من أن مجموع مخرجات *CNN* هو 1. وبسبب هذا، فإن عمليات *softmax* مفيدة لتوسيع نطاق مخرجات النموذج إلى احتمالات.

3. Pooling Layers:

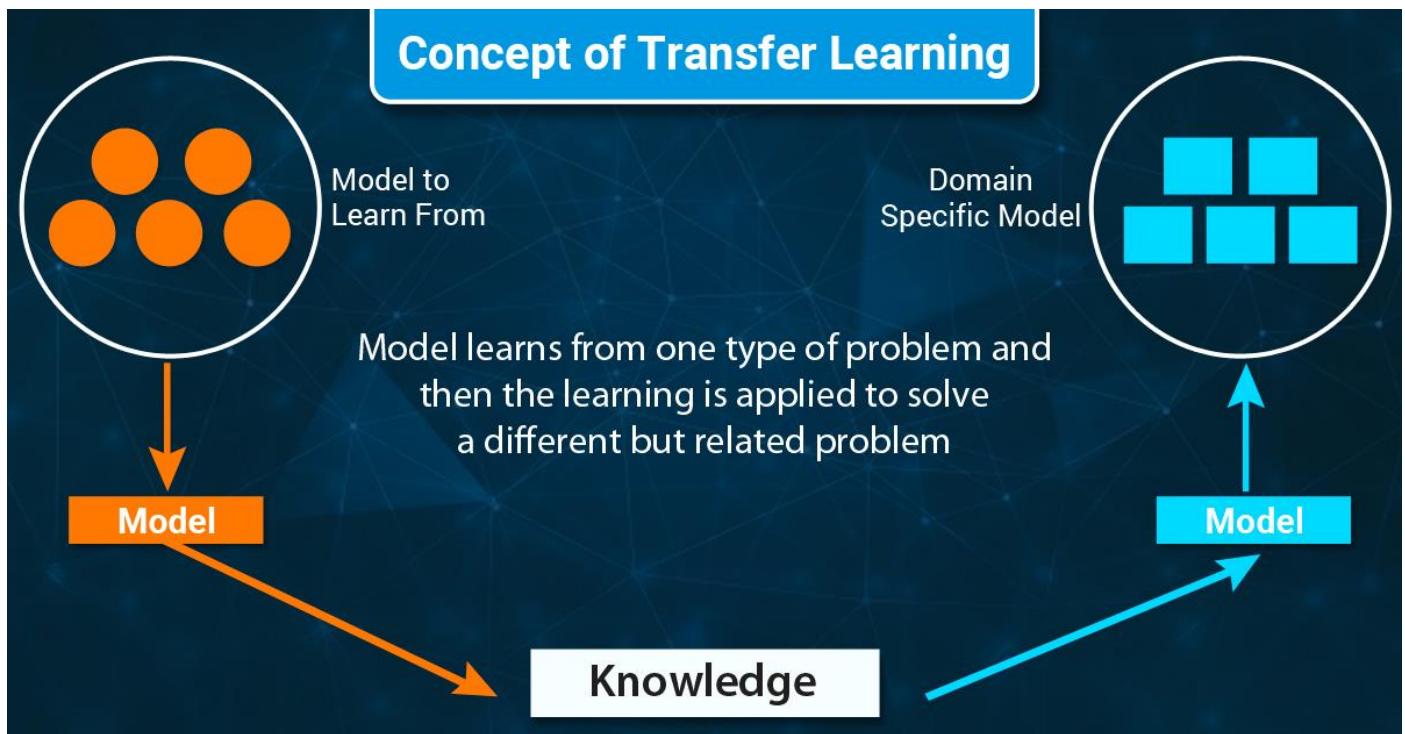
هناك العديد من أنواع طبقات التجميع في بنى الشبكات العصبية التلفيفية المختلفة، ولكن جميعها تهدف إلى تقليل المدى المكاني للشبكة تدريجياً، مما يقلل من المعلومات والحساب الكلي للشبكة.

تتطبع عملية Max – Pooling تحديد حجم النواة وخطوة الطول أثناء تصميم بنية الشبكة. بمجرد التحديد، تقوم العملية بتمرير النواة بخطوة محددة فوق المدخلات مع تحديد القيمة الأكبر فقط في كل شريحة نواة من الإدخال لإعطاء قيمة للإخراج.

في بنية Tiny VGG، تستخدم طبقات التجميع نواة 2×2 وخطوة 2. ينتج عن هذه العملية بهذه المواصفات تجاهل 75% من عمليات التشويش. من خلال تجاهل العديد من القيم، يكون Tiny VGG أكثر كفاءة من الناحية الحسابية ويتجنب الإفراط في التجهيز.

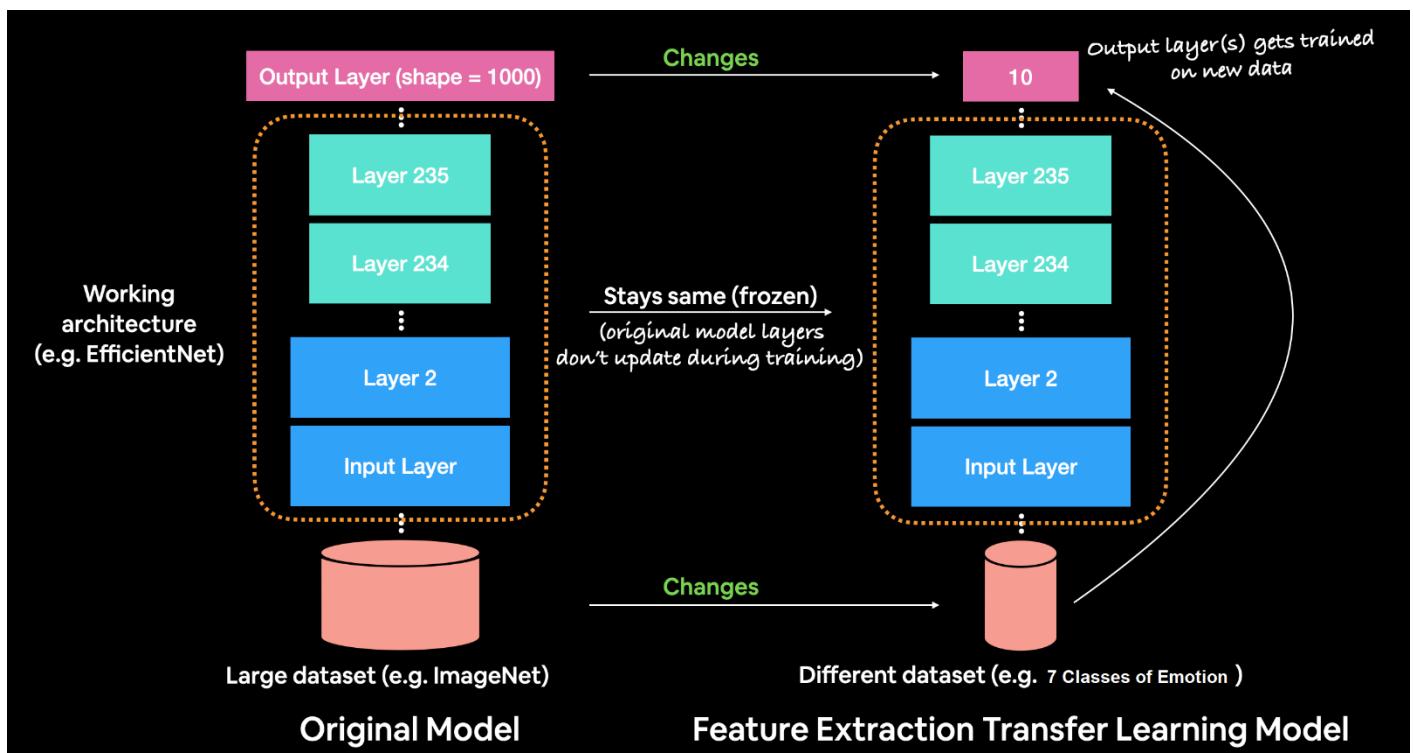
4. Flatten Layers

تقوم هذه الطبقة بتحويل طبقة ثلاثية الأبعاد في الشبكة إلى متوجه أحادي البعد لتناسب مدخلات طبقة متصلة بالكامل من أجل التصنيف. على سبيل المثال، سيتم تحويل موتر $5 \times 5 \times 2$ إلى متوجه بحجم 50. استخرجت الطبقات التلفيفية السابقة للشبكة الميزات من صورة الإدخال، ولكن حان الوقت الآن لتصنيف الميزات. نستخدم وظيفة softmax لتصنيف هذه الميزات، والتي تتطلب إدخالاً أحادي البعد. هذا هو السبب في أن طبقة ال(Flatten) ضرورية.



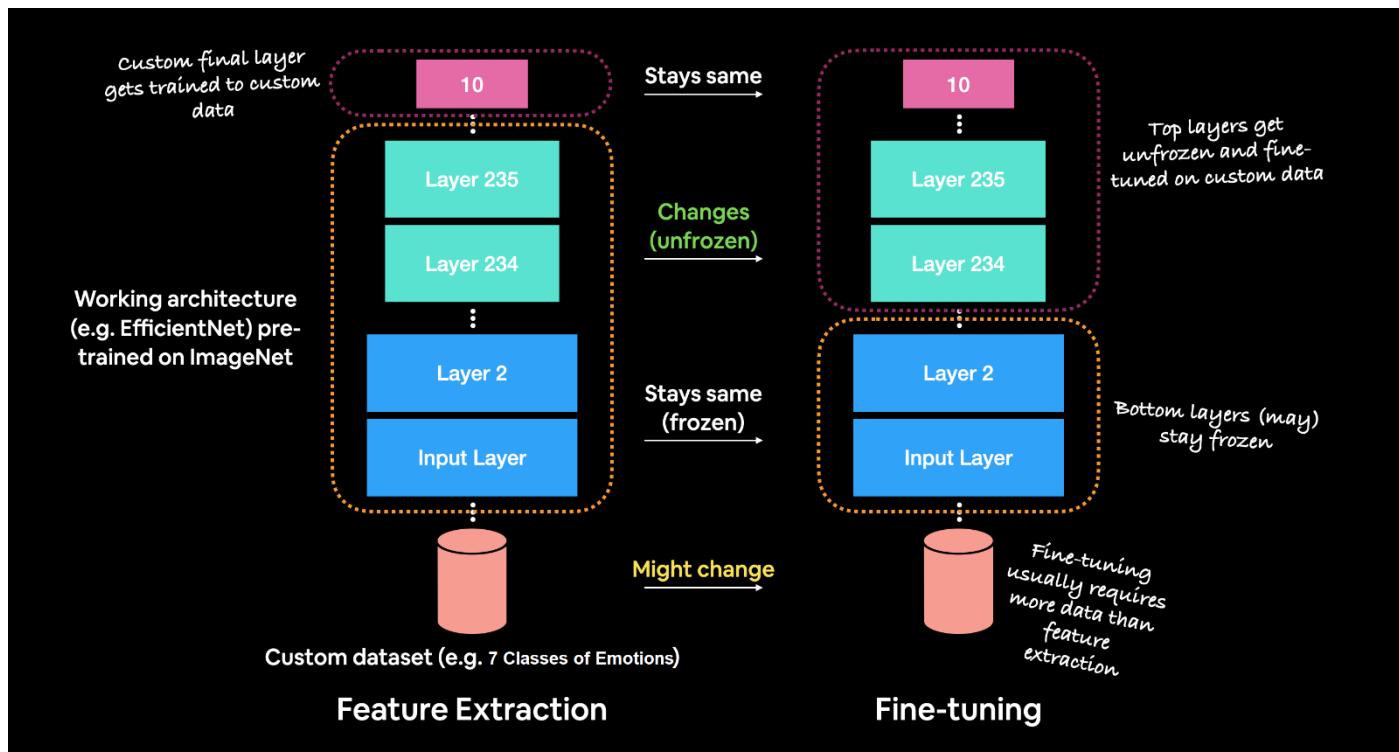
8. Transfer Learning:

هو مشكلة بحثية في التعلم الآلي (ML) تركز على تخزين المعرفة المكتسبة أثناء حل مشكلة واحدة وتطبيقاتها على مشكلة مختلفة ولكنها ذات صلة. على سبيل المثال، يمكن تطبيق المعرفة المكتسبة أثناء تعلم التعرف على السيارات عند محاولة التعرف على الشاحنات. هذا المجال من البحث له علاقة ما بالتاريخ الطويل للأدب النفسي حول نقل التعلم، على الرغم من أن الروابط العملية بين المجالين محدودة. من وجهة النظر العلمية، فإن إعادة استخدام أو نقل المعلومات من المهام التي تم تعلمها سابقاً لتعلم المهام الجديدة لديها القدرة على تحسين كفاءة عينة عامل التعلم المعزز بشكل كبير.



8. Feature Extraction in Transfer learning:

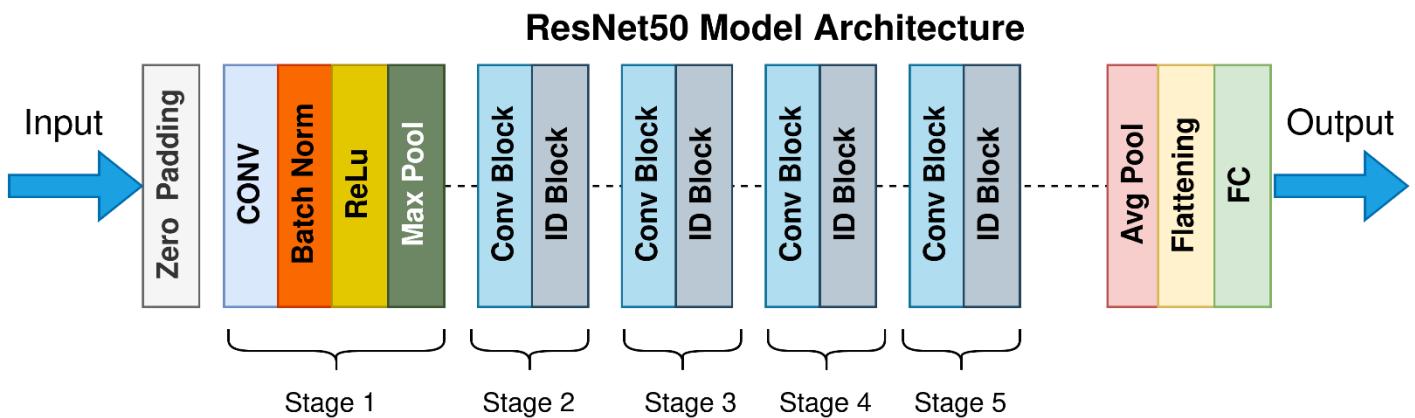
في الـ (*Transfer learning*) إن استخلاص السمات هو عبارة عنأخذ أوزان شبكة مدربة مسبقاً (Resnet50, EfficientNetB0) مثلاً، وضبط مخرجاتها لتكون ملائمة لمشكلتنا (التناسب مع 7 صفوف من المشاعر مثلاً) واستخدامه الأكبر يأتي في حين نجد الحصول على دقة في نموذجنا عند عدم وجود بيانات كافية (وهذا ما تم العمل عليه في بناء نماذجنا).



9. Fine Tuning in Transfer learning:

في الـ (*Transfer learning*) يكون الضبط الدقيق بإلغاء تجميد أوزان النموذج المدربة مسبقاً من نموذج آخر وتعديلها لتناسب مشكلتنا بشكل أفضل.

بالنسبة لاستخراج السمات في الـ (*Transfer learning*), يمكن فقط تدريب الطبقات (1 – 3) العليا لنموذج مدرب مسبقاً بمجموعات بياناتنا الخاصة، بينما في الضبط الدقيق في الـ (*Transfer learning*), يمكنك تدريب (1 – 3 +) طبقات من نموذج مدرب مسبقاً (حيث تشير علامة "+" إلى أنه يمكن تدريب العديد من الطبقات أو جميعها).



10. Resnet50:

هي عبارة عن شبكة عصبية تلفيفية بعمق 50 طبقة. يمكنك تحميل نسخة مسبقة التدريب من الشبكة تم تدريبيها على أكثر من مليون صورة من قاعدة بيانات (ImageNet). يمكن للشبكة التي تم اختبارها مسبقاً تصنيف الصور إلى 1000 فئة كائن، مثل لوحة المفاتيح والماوس والقلم الرصاص والعديد من الحيوانات. نتيجة لذلك، تعلمت الشبكة تمثيلات غنية بالميزات لمجموعة واسعة من الصور. يبلغ حجم إدخال الصور للشبكة 224×224 .

11. EfficientNet – B0:

هي عبارة عن شبكة عصبية تلفيفية تم تدريبيها على أكثر من مليون صورة من قاعدة بيانات (ImageNet). يمكن للشبكة تصنيف الصور إلى 1000 فئة كائن، مثل لوحة المفاتيح والماوس والقلم الرصاص والعديد من الحيوانات. نتيجة لذلك، تعلمت الشبكة تمثيلات غنية بالميزات لمجموعة واسعة من الصور. يبلغ حجم إدخال الصور للشبكة 224×224 . عند التعامل معها لا تحتاج لعمل (Normalization) لدخلها بسبب وجود طبقة (Normalization) من أصل الشبكة.

الفصل الخامس

تقرير عن النتائج والتنفيذ

الملخص: يقدم هذا التقرير محاولتنا لبناء العديد من نماذج التعلم العميق من أجل التعرف على مشاعر الإنسان من تعبير وجهه عن طريق صور الوجه، إن مجموعة البيانات التي قمنا بالاعتماد عليها في نماذجنا هي (*FER – 2013*)، ولكن لم نقم بالاختبار عليها كاملاً دفعة واحدة دائماً، فقد قمنا بتقسيمها حسب احتياجنا، من أجل أن نقوم باختباراتنا بسرعة، ونستطيع معرفة ما يمكن أن يجعل نماذجنا أفضل أو أسوء.

ومن أجل تقييم نماذج التعلم العميق. فسوف نقوم بعرض ما يلي:

1. تقييم النموذج.
2. عرض تلخيص النموذج.
3. عرض منحنى الضياع ومنحنى الدقة (*loss & accuracy curves*).
4. عرض مصفوفة الاختلاط (*confusion matrix*).
5. عرض تقرير التصنيف (*classification report*).

وتم تدريب كل نماذج التعلم العميق على نظام (*Google Colab*) باستخدام (GPU) وحدة معالجة الرسوميات، بسبب أن سرعة تدريب النموذج على وحدة معالجة الرسوميات من الممكن أن يصل إلى 30 ضعف سرعة تدريبيه على وحدة المعالجة المركزية. ووحدة المعالجة التي قمنا باستعمالها هي:

```
+-----+
| NVIDIA-SMI 460.32.03      Driver Version: 460.32.03      CUDA Version: 11.2 |
+-----+
| GPU  Name      Persistence-M| Bus-Id     Disp.A  | Volatile Uncorr. ECC | | | | |
| Fan  Temp  Perf  Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M. |
|          |          |          |          |          |          | MIG M. |
+-----+
| 0  Tesla K80          Off  | 00000000:00:04.0 Off |                  0 | | | | |
| N/A   54C   P8    31W / 149W |      0MiB / 11441MiB |      0%     Default |
|          |          |          |          |          |          | N/A |
+-----+
+-----+
| Processes:
| GPU  GI  CI      PID  Type  Process name          GPU Memory |
|           ID  ID
+-----+
| No running processes found
+-----+
```

يقسم التنفيذ لدينا إلى 5 ملفات:

- الملف الأول: تدريب نماذج تعلم عميق باستخدام الشبكات العصبية التأفيافية من أجل تحديد مشاعر الحزن والسعادة.
- الملف الثاني: تدريب نماذج تعلم عميق باستخدام الشبكات العصبية التأفيافية من أجل تحديد مشاعر.
- الملف الثالث: استخدام التعلم العميق في استخراج السمات.
- الملف الرابع: استخدام التعلم العميق في الضبط الدقيق.
- الملف الخامس: النموذج النهائي لشبكة عصبية تأفيافية من أجل التعرف على المشاعر السبعة عند الإنسان.

الملف الأول

قمنا باعتماد مجموعة صور من أصل مجموعة البيانات كاملة، وهما السعادة والحزن.

الحزن	السعادة	مجموعات التدريب
صورة 4830	صورة 7215	مجموعات الاختبار
صورة 1247	صورة 1774	

النموذج الأول:

1. تقييم النموذج:

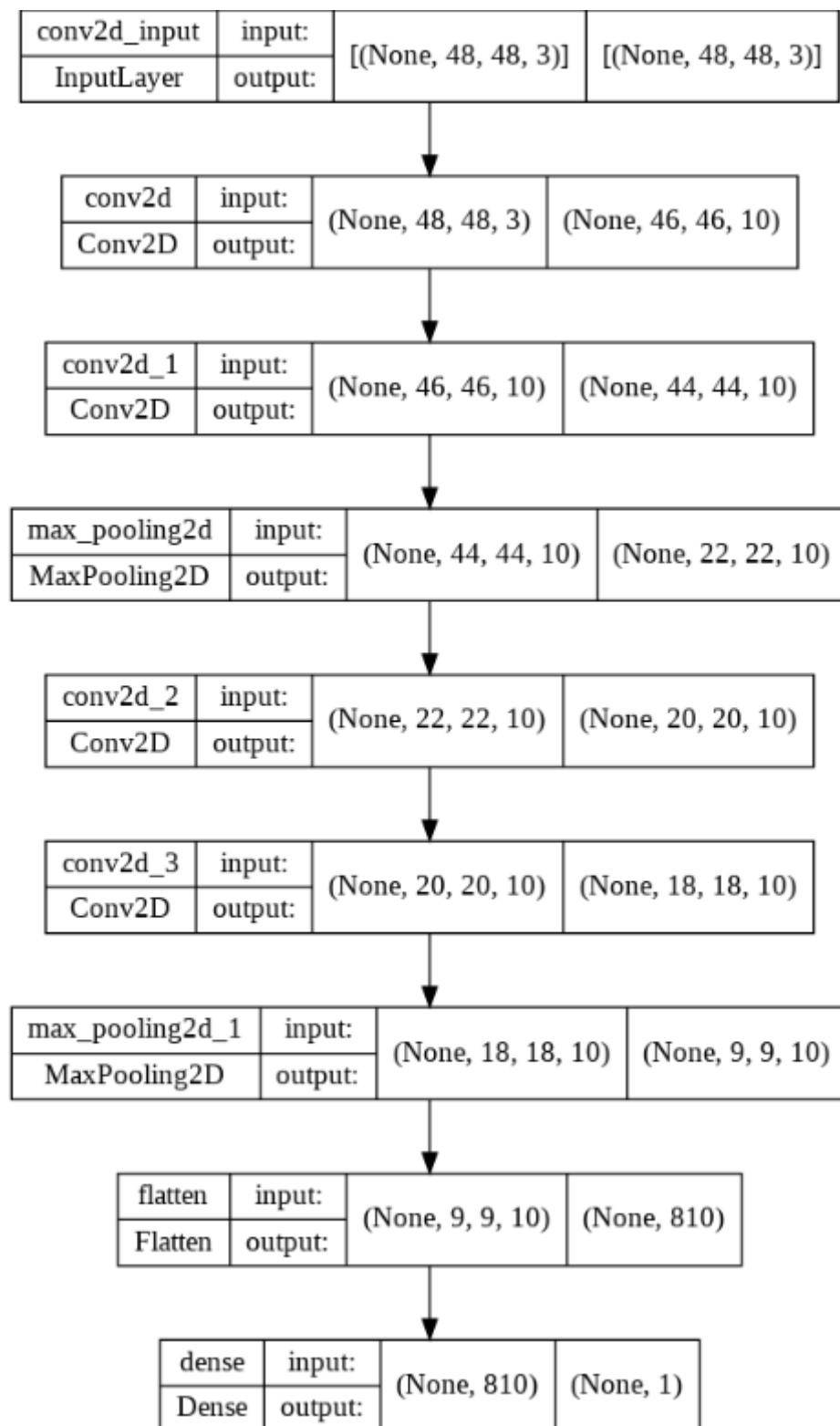
```
[ ] # Evaluate the loaded model
model_1.evaluate(train_data), model_1.evaluate(valid_data)

377/377 [=====] - 6s 16ms/step - loss: 0.3087 - accuracy: 0.8584
95/95 [=====] - 1s 15ms/step - loss: 0.3652 - accuracy: 0.8375
([0.30871447920799255, 0.858364462852478],
 [0.3651982247829437, 0.8374710083007812])
```

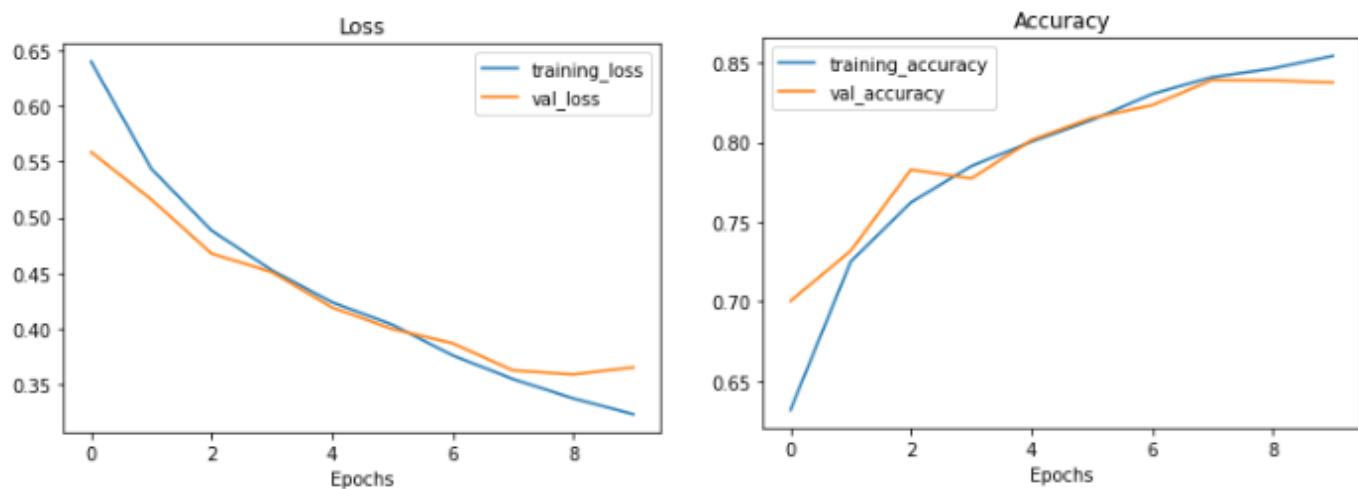
وقد تم تدريب النموذج لـ 10 دورات (epochs)

```
Found 12045 images belonging to 2 classes.
Found 3021 images belonging to 2 classes.
Epoch 1/10
377/377 [=====] - 22s 22ms/step - loss: 0.6399 - accuracy: 0.6320 - val_loss: 0.5586 - val_accuracy: 0.7004
Epoch 2/10
377/377 [=====] - 8s 21ms/step - loss: 0.5437 - accuracy: 0.7251 - val_loss: 0.5160 - val_accuracy: 0.7319
Epoch 3/10
377/377 [=====] - 8s 21ms/step - loss: 0.4881 - accuracy: 0.7623 - val_loss: 0.4675 - val_accuracy: 0.7829
Epoch 4/10
377/377 [=====] - 8s 20ms/step - loss: 0.4525 - accuracy: 0.7851 - val_loss: 0.4508 - val_accuracy: 0.7772
Epoch 5/10
377/377 [=====] - 8s 21ms/step - loss: 0.4236 - accuracy: 0.8003 - val_loss: 0.4190 - val_accuracy: 0.8014
Epoch 6/10
377/377 [=====] - 8s 20ms/step - loss: 0.4036 - accuracy: 0.8139 - val_loss: 0.3997 - val_accuracy: 0.8153
Epoch 7/10
377/377 [=====] - 8s 20ms/step - loss: 0.3759 - accuracy: 0.8303 - val_loss: 0.3869 - val_accuracy: 0.8232
Epoch 8/10
377/377 [=====] - 8s 20ms/step - loss: 0.3547 - accuracy: 0.8409 - val_loss: 0.3624 - val_accuracy: 0.8391
Epoch 9/10
377/377 [=====] - 8s 22ms/step - loss: 0.3373 - accuracy: 0.8462 - val_loss: 0.3588 - val_accuracy: 0.8388
Epoch 10/10
377/377 [=====] - 8s 20ms/step - loss: 0.3231 - accuracy: 0.8543 - val_loss: 0.3652 - val_accuracy: 0.8375
```

٢. تلخيص النموذج.



3. منحنى الضياع ومنحنى الدقة (loss & accuracy curves)



النموذج الثاني:

قمنا بنسخ نموذج موقع (*Tensorflow Playground*)

1. تقييم النموذج:

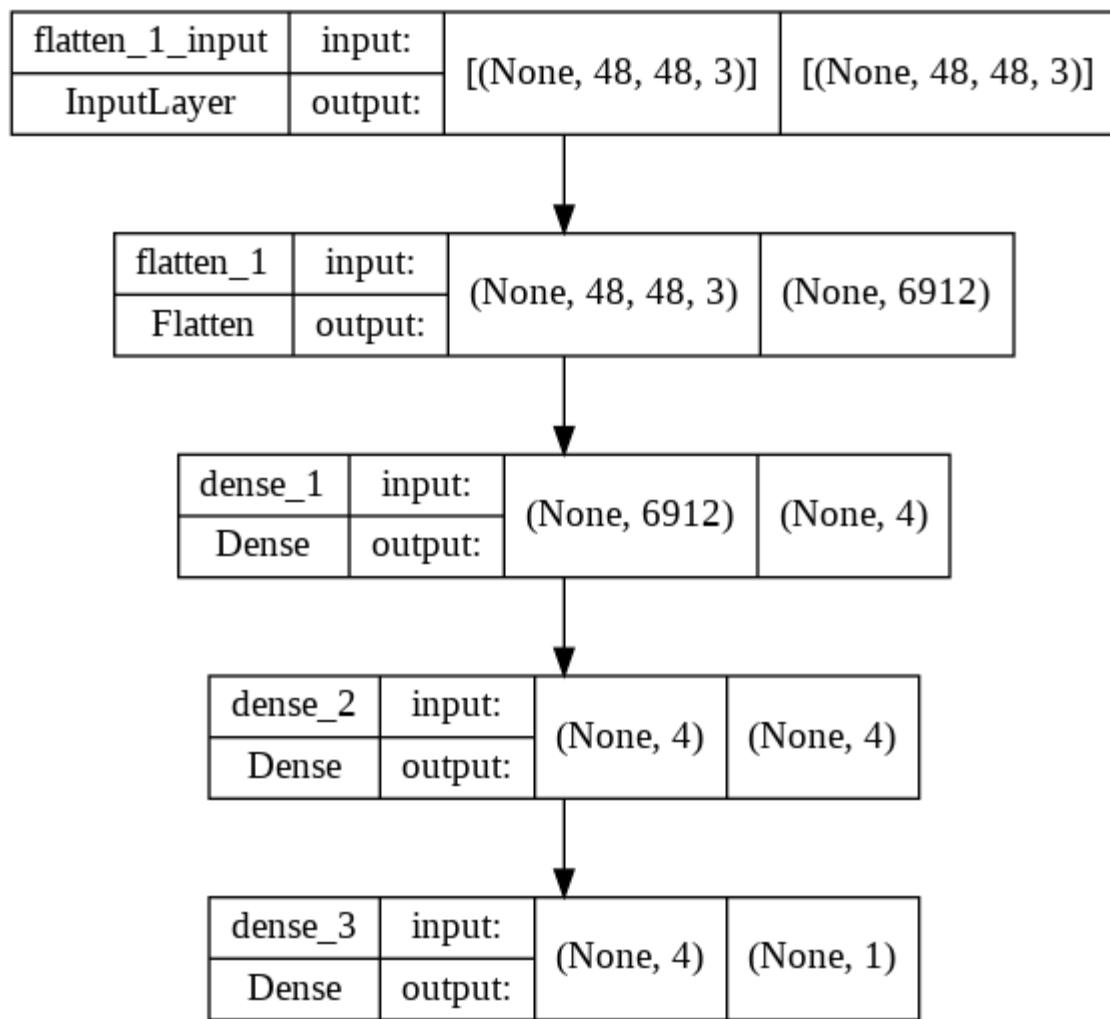
```
[ ] # Evaluate the loaded model
model_2.evaluate(train_data), model_2.evaluate(valid_data)

377/377 [=====] - 5s 14ms/step - loss: 0.5562 - accuracy: 0.7230
95/95 [=====] - 1s 14ms/step - loss: 0.5584 - accuracy: 0.7219
([0.5561778545379639, 0.722955584526062],
 [0.5583581924438477, 0.7219463586807251])
```

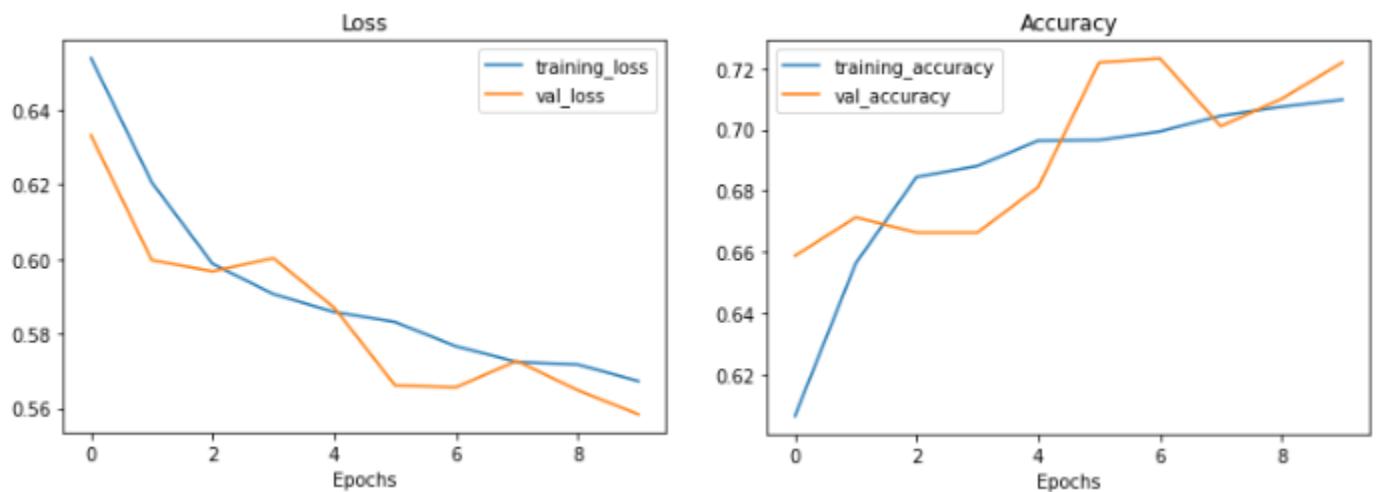
وقد تم تدريب النموذج لـ 10 دورات (epochs)

```
Epoch 1/10
377/377 [=====] - 7s 18ms/step - loss: 0.6540 - accuracy: 0.6064 - val_loss: 0.6333 - val_accuracy: 0.6587
Epoch 2/10
377/377 [=====] - 7s 18ms/step - loss: 0.6206 - accuracy: 0.6563 - val_loss: 0.5997 - val_accuracy: 0.6713
Epoch 3/10
377/377 [=====] - 7s 18ms/step - loss: 0.5987 - accuracy: 0.6845 - val_loss: 0.5967 - val_accuracy: 0.6663
Epoch 4/10
377/377 [=====] - 7s 18ms/step - loss: 0.5906 - accuracy: 0.6881 - val_loss: 0.6003 - val_accuracy: 0.6663
Epoch 5/10
377/377 [=====] - 7s 18ms/step - loss: 0.5858 - accuracy: 0.6964 - val_loss: 0.5870 - val_accuracy: 0.6812
Epoch 6/10
377/377 [=====] - 7s 19ms/step - loss: 0.5831 - accuracy: 0.6966 - val_loss: 0.5661 - val_accuracy: 0.7219
Epoch 7/10
377/377 [=====] - 7s 18ms/step - loss: 0.5766 - accuracy: 0.6994 - val_loss: 0.5656 - val_accuracy: 0.7233
Epoch 8/10
377/377 [=====] - 7s 18ms/step - loss: 0.5724 - accuracy: 0.7045 - val_loss: 0.5727 - val_accuracy: 0.7011
Epoch 9/10
377/377 [=====] - 7s 18ms/step - loss: 0.5717 - accuracy: 0.7074 - val_loss: 0.5648 - val_accuracy: 0.7100
Epoch 10/10
377/377 [=====] - 7s 18ms/step - loss: 0.5673 - accuracy: 0.7098 - val_loss: 0.5584 - val_accuracy: 0.7219
```

2. تلخيص النموذج:



3. منحنى الضياع ومنحنى الدقة (loss & accuracy curves)



النموذج الثالث:

1. تقييم النموذج:

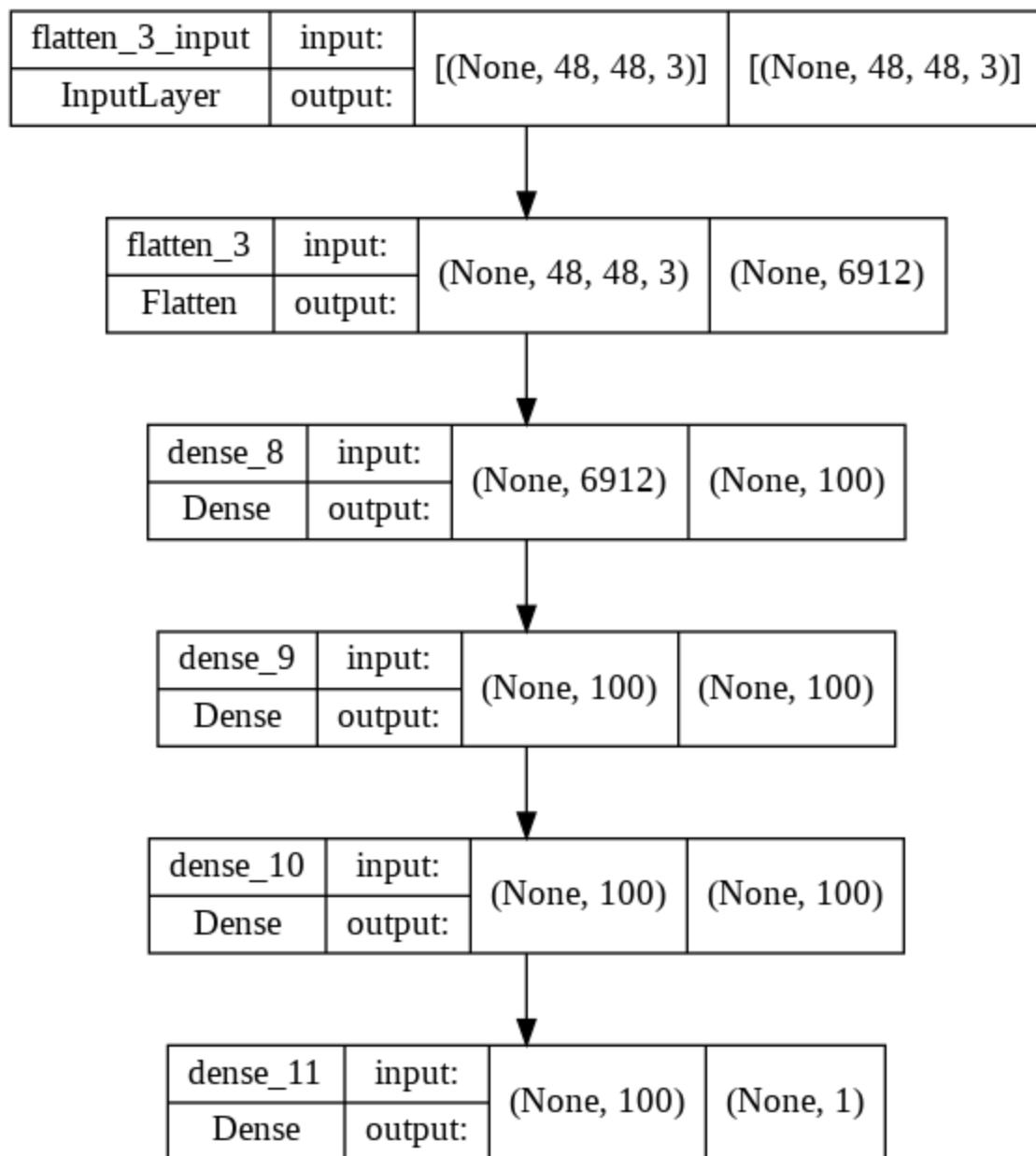
```
[ ] # Evaluate the loaded model
model_3.evaluate(train_data), model_3.evaluate(valid_data)

377/377 [=====] - 5s 14ms/step - loss: 0.5592 - accuracy: 0.7279
95/95 [=====] - 1s 14ms/step - loss: 0.5599 - accuracy: 0.7279
([0.559227466583252, 0.7279369235038757],
 [0.5598790645599365, 0.7279046773910522])
```

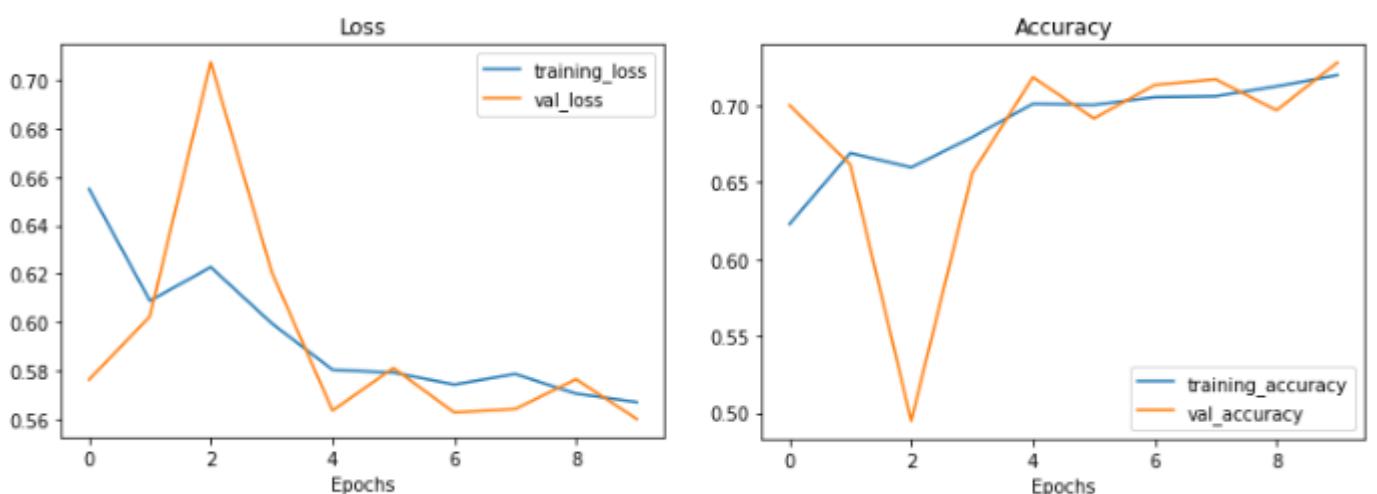
وقد تم تدريب النموذج لـ 10 دورات (epochs):

```
Epoch 1/10
377/377 [=====] - 9s 21ms/step - loss: 0.6551 - accuracy: 0.6227 - val_loss: 0.5762 - val_accuracy: 0.7001
Epoch 2/10
377/377 [=====] - 7s 18ms/step - loss: 0.6088 - accuracy: 0.6690 - val_loss: 0.6023 - val_accuracy: 0.6614
Epoch 3/10
377/377 [=====] - 7s 18ms/step - loss: 0.6227 - accuracy: 0.6596 - val_loss: 0.7074 - val_accuracy: 0.4949
Epoch 4/10
377/377 [=====] - 7s 18ms/step - loss: 0.5995 - accuracy: 0.6793 - val_loss: 0.6208 - val_accuracy: 0.6557
Epoch 5/10
377/377 [=====] - 7s 18ms/step - loss: 0.5803 - accuracy: 0.7010 - val_loss: 0.5634 - val_accuracy: 0.7183
Epoch 6/10
377/377 [=====] - 7s 18ms/step - loss: 0.5791 - accuracy: 0.7004 - val_loss: 0.5810 - val_accuracy: 0.6915
Epoch 7/10
377/377 [=====] - 7s 18ms/step - loss: 0.5742 - accuracy: 0.7054 - val_loss: 0.5626 - val_accuracy: 0.7133
Epoch 8/10
377/377 [=====] - 7s 18ms/step - loss: 0.5785 - accuracy: 0.7059 - val_loss: 0.5640 - val_accuracy: 0.7170
Epoch 9/10
377/377 [=====] - 7s 18ms/step - loss: 0.5704 - accuracy: 0.7124 - val_loss: 0.5764 - val_accuracy: 0.6968
Epoch 10/10
377/377 [=====] - 7s 18ms/step - loss: 0.5669 - accuracy: 0.7196 - val_loss: 0.5599 - val_accuracy: 0.7279
```

2. تلخيص النموذج:



3. منحنى الضياع ومنحنى الدقة (loss & accuracy curves)



النموذج الرابع:

1. تقييم النموذج:

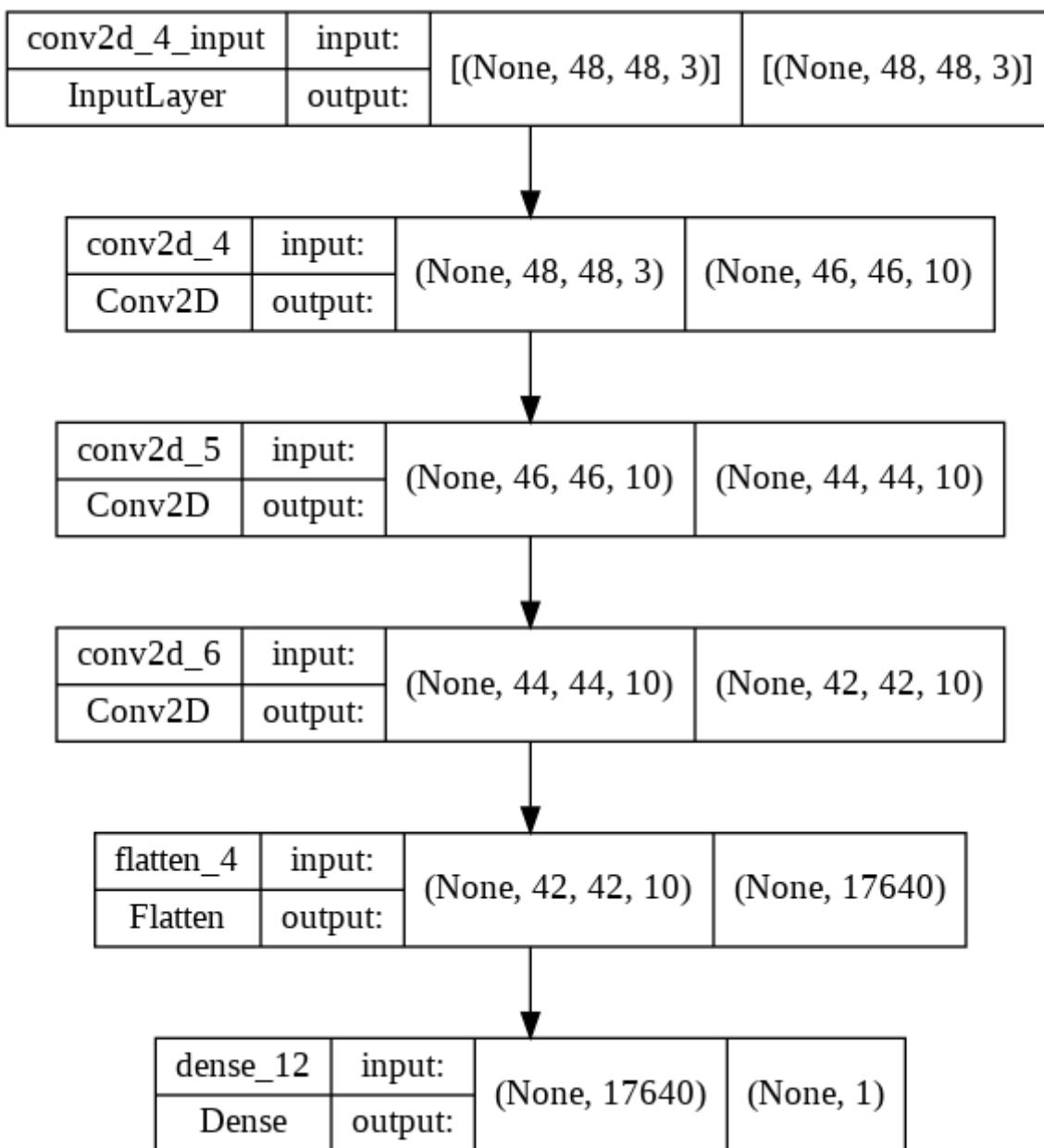
```
[ ] # Evaluate the loaded model
model_4.evaluate(train_data), model_4.evaluate(test_data)

377/377 [=====] - 6s 15ms/step - loss: 0.1872 - accuracy: 0.9359
95/95 [=====] - 1s 15ms/step - loss: 0.5033 - accuracy: 0.7809
([0.1871885061264038, 0.9359070062637329],
 [0.5032828450202942, 0.7808672785758972])
```

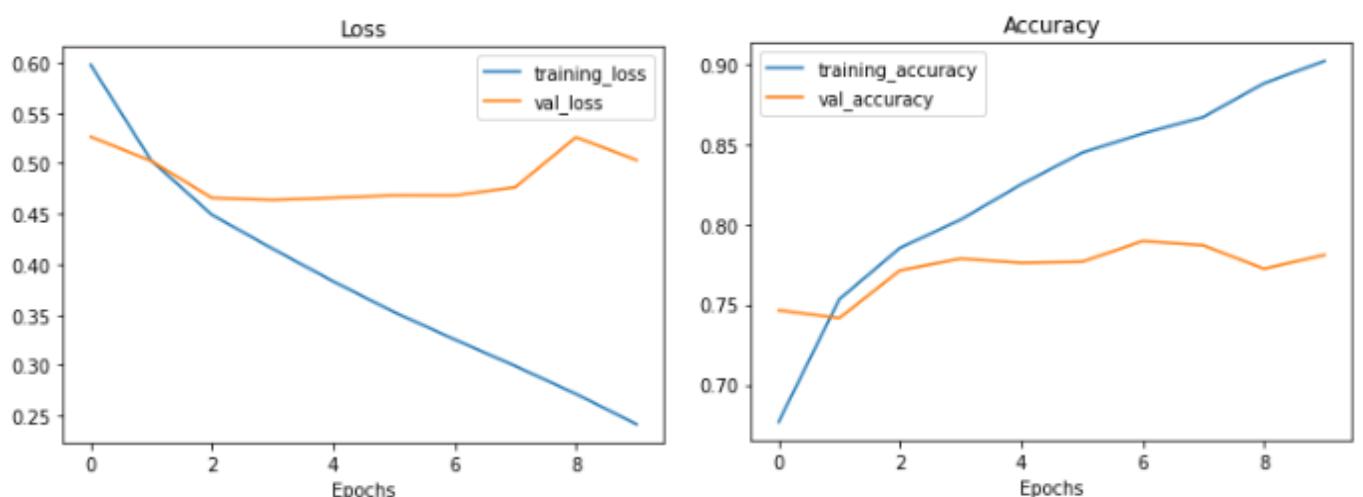
وقد تم تدريب النموذج لـ 10 دورات (epochs)

```
Epoch 1/10
377/377 [=====] - 12s 28ms/step - loss: 0.5978 - accuracy: 0.6769 - val_loss: 0.5265 - val_accuracy: 0.7464
Epoch 2/10
377/377 [=====] - 8s 21ms/step - loss: 0.5027 - accuracy: 0.7533 - val_loss: 0.5024 - val_accuracy: 0.7415
Epoch 3/10
377/377 [=====] - 8s 21ms/step - loss: 0.4490 - accuracy: 0.7852 - val_loss: 0.4658 - val_accuracy: 0.7713
Epoch 4/10
377/377 [=====] - 8s 21ms/step - loss: 0.4154 - accuracy: 0.8028 - val_loss: 0.4637 - val_accuracy: 0.7786
Epoch 5/10
377/377 [=====] - 8s 21ms/step - loss: 0.3831 - accuracy: 0.8248 - val_loss: 0.4658 - val_accuracy: 0.7759
Epoch 6/10
377/377 [=====] - 8s 21ms/step - loss: 0.3526 - accuracy: 0.8445 - val_loss: 0.4683 - val_accuracy: 0.7766
Epoch 7/10
377/377 [=====] - 8s 21ms/step - loss: 0.3255 - accuracy: 0.8565 - val_loss: 0.4683 - val_accuracy: 0.7895
Epoch 8/10
377/377 [=====] - 8s 22ms/step - loss: 0.2991 - accuracy: 0.8667 - val_loss: 0.4764 - val_accuracy: 0.7868
Epoch 9/10
377/377 [=====] - 8s 20ms/step - loss: 0.2715 - accuracy: 0.8878 - val_loss: 0.5260 - val_accuracy: 0.7723
Epoch 10/10
377/377 [=====] - 8s 20ms/step - loss: 0.2414 - accuracy: 0.9017 - val_loss: 0.5033 - val_accuracy: 0.7809
```

عرض تلخيص النموذج:



3. منحنى الضياع ومنحنى الدقة (loss & accuracy curves)



نلاحظ أن هناك فرق كبير بين منحنى مجموعات التدريب ومجموعات التحقق، وبالحالة المثالية، يجب أن يكونا متقاربين قدر الإمكان، وهذا يخبرنا أن نموذجنا من الممكن أن يكون **(overfitting)**.

النموذج الخامس:

1. تقييم النموذج:

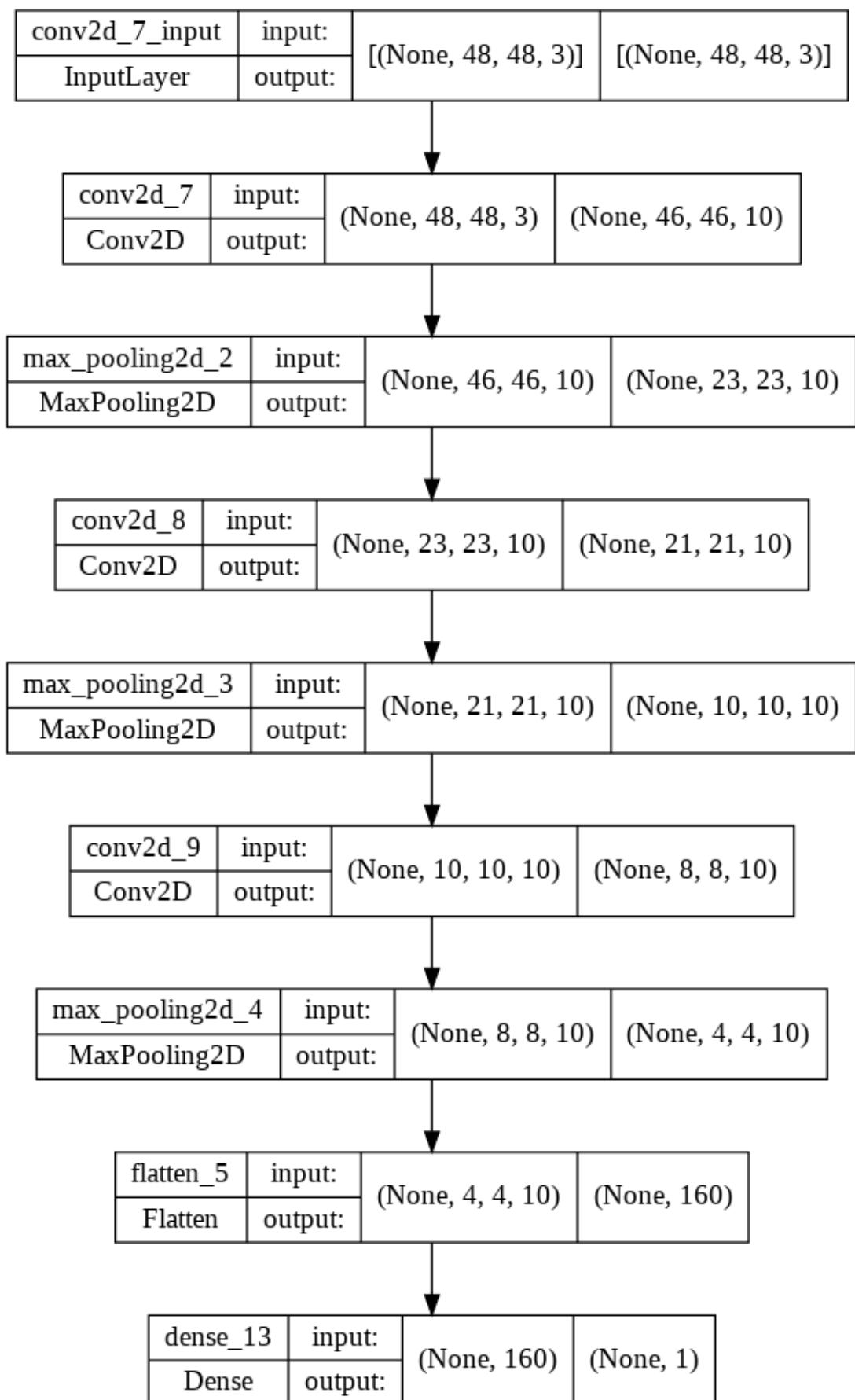
```
[ ] # Evaluate the loaded model
model_5.evaluate(train_data), model_5.evaluate(valid_data)
```

```
377/377 [=====] - 5s 14ms/step - loss: 0.3656 - accuracy: 0.8363
95/95 [=====] - 1s 14ms/step - loss: 0.4006 - accuracy: 0.8160
([0.3655555844306946, 0.8362806439399719],
 [0.4006417691707611, 0.8159549832344055])
```

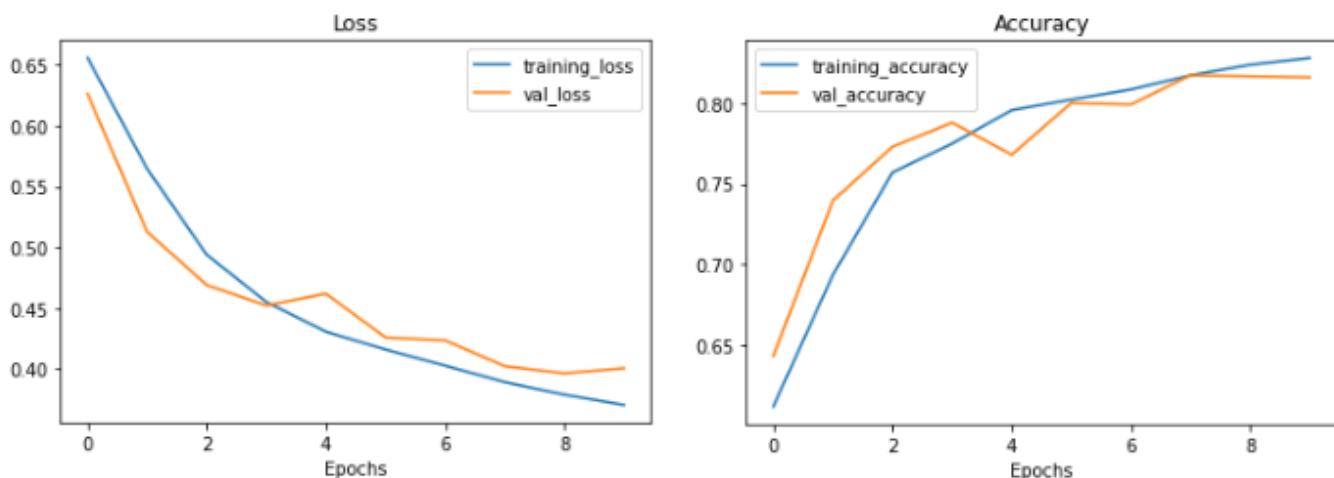
وقد تم تدريب النموذج لـ **10 دورات (epochs)**:

```
Epoch 1/10
377/377 [=====] - 12s 27ms/step - loss: 0.6558 - accuracy: 0.6119 - val_loss: 0.6261 - val_accuracy: 0.6432
Epoch 2/10
377/377 [=====] - 7s 19ms/step - loss: 0.5645 - accuracy: 0.6935 - val_loss: 0.5126 - val_accuracy: 0.7395
Epoch 3/10
377/377 [=====] - 7s 19ms/step - loss: 0.4939 - accuracy: 0.7570 - val_loss: 0.4690 - val_accuracy: 0.7729
Epoch 4/10
377/377 [=====] - 7s 19ms/step - loss: 0.4551 - accuracy: 0.7748 - val_loss: 0.4522 - val_accuracy: 0.7878
Epoch 5/10
377/377 [=====] - 7s 19ms/step - loss: 0.4306 - accuracy: 0.7956 - val_loss: 0.4619 - val_accuracy: 0.7680
Epoch 6/10
377/377 [=====] - 7s 19ms/step - loss: 0.4160 - accuracy: 0.8022 - val_loss: 0.4259 - val_accuracy: 0.8001
Epoch 7/10
377/377 [=====] - 7s 19ms/step - loss: 0.4028 - accuracy: 0.8084 - val_loss: 0.4235 - val_accuracy: 0.7991
Epoch 8/10
377/377 [=====] - 7s 19ms/step - loss: 0.3892 - accuracy: 0.8172 - val_loss: 0.4023 - val_accuracy: 0.8173
Epoch 9/10
377/377 [=====] - 7s 19ms/step - loss: 0.3789 - accuracy: 0.8237 - val_loss: 0.3965 - val_accuracy: 0.8166
Epoch 10/10
377/377 [=====] - 7s 19ms/step - loss: 0.3705 - accuracy: 0.8280 - val_loss: 0.4006 - val_accuracy: 0.8160
```

عرض تلخيص النموذج:



3. منحنى الضياع ومنحنى الدقة :(*loss & accuracy curves*)



في هذا النموذج قمنا بزيادة طبقات الشبكة العصبية.

النموذج السادس:

1. تقييم النموذج:

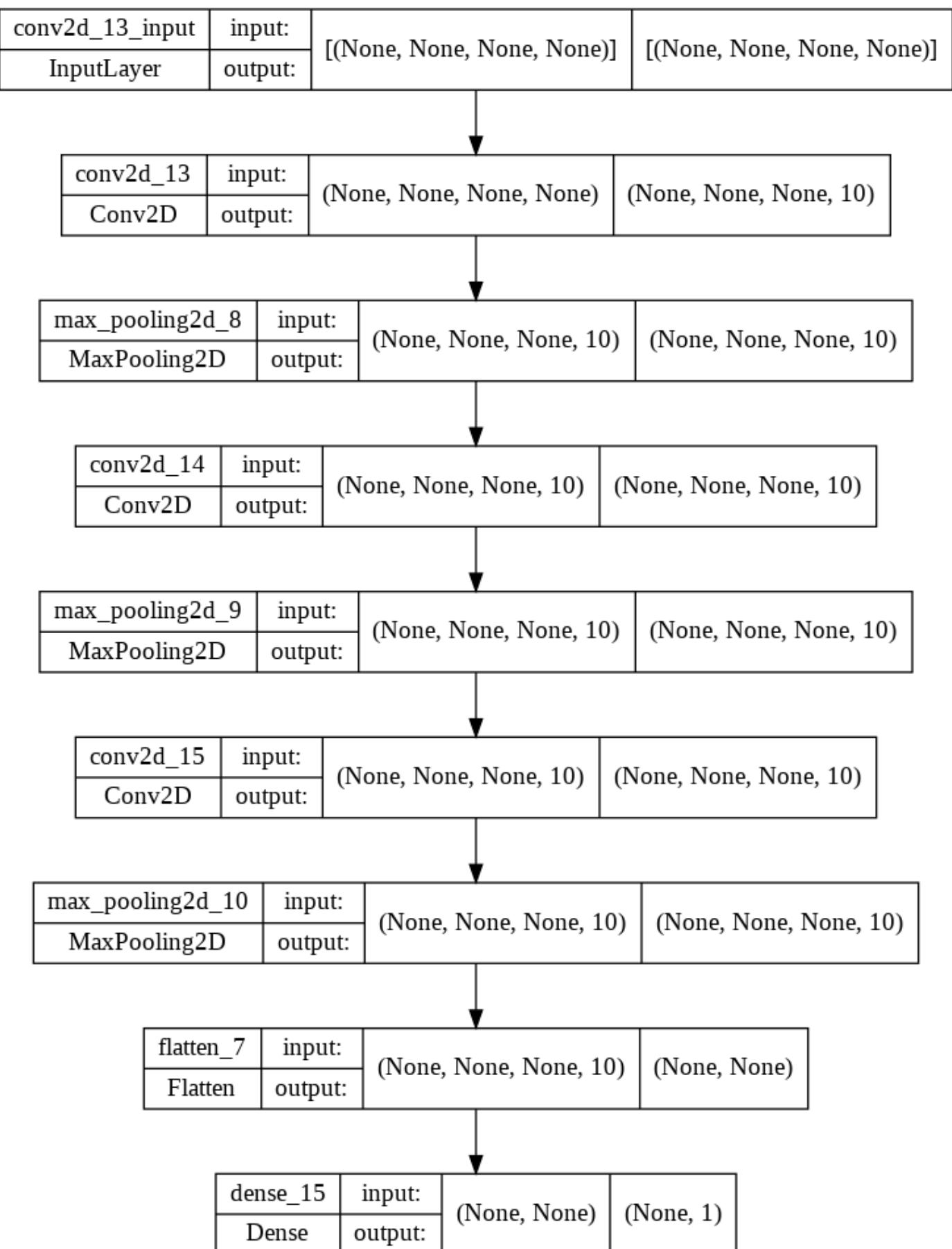
```
[ ] model_6.evaluate(train_data), model_6.evaluate(test_data)
```

```
377/377 [=====] - 6s 15ms/step - loss: 0.6391 - accuracy: 0.6568
95/95 [=====] - 1s 14ms/step - loss: 0.6356 - accuracy: 0.6640
([0.6390643119812012, 0.6567870378494263],
 [0.6355599761009216, 0.6640185117721558])
```

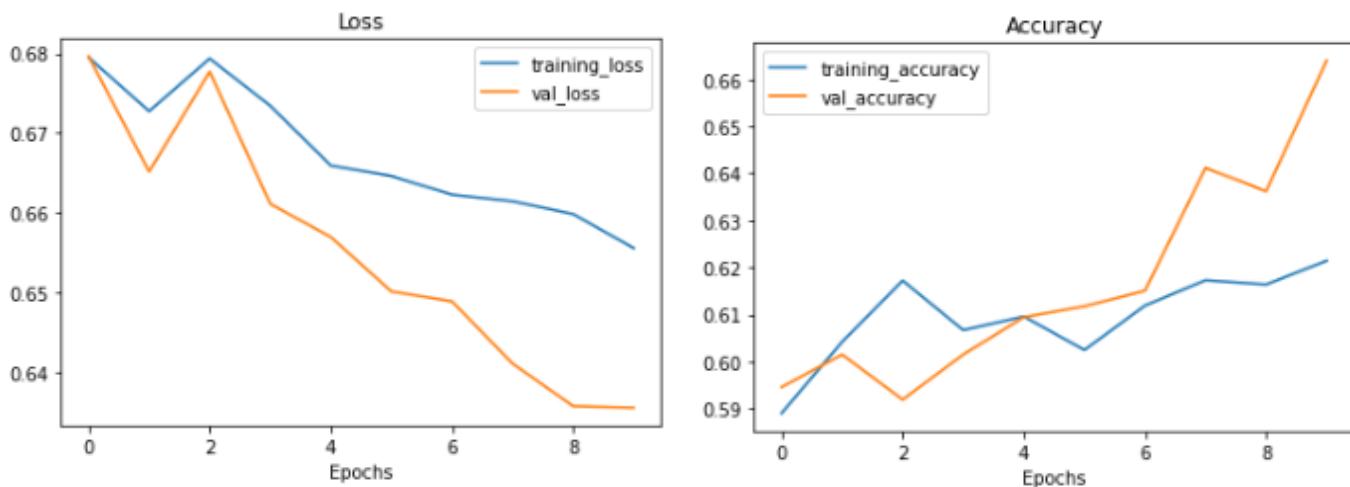
وقد تم تدريب النموذج لـ 10 دورات (*epochs*)

```
Epoch 1/10
377/377 [=====] - 23s 58ms/step - loss: 0.6795 - accuracy: 0.5890 - val_loss: 0.6796 - val_accuracy: 0.5945
Epoch 2/10
377/377 [=====] - 18s 48ms/step - loss: 0.6728 - accuracy: 0.6042 - val_loss: 0.6652 - val_accuracy: 0.6015
Epoch 3/10
377/377 [=====] - 18s 48ms/step - loss: 0.6794 - accuracy: 0.6172 - val_loss: 0.6777 - val_accuracy: 0.5919
Epoch 4/10
377/377 [=====] - 18s 48ms/step - loss: 0.6734 - accuracy: 0.6066 - val_loss: 0.6611 - val_accuracy: 0.6015
Epoch 5/10
377/377 [=====] - 18s 49ms/step - loss: 0.6659 - accuracy: 0.6095 - val_loss: 0.6570 - val_accuracy: 0.6094
Epoch 6/10
377/377 [=====] - 19s 50ms/step - loss: 0.6646 - accuracy: 0.6024 - val_loss: 0.6501 - val_accuracy: 0.6117
Epoch 7/10
377/377 [=====] - 19s 50ms/step - loss: 0.6623 - accuracy: 0.6119 - val_loss: 0.6489 - val_accuracy: 0.6150
Epoch 8/10
377/377 [=====] - 18s 48ms/step - loss: 0.6615 - accuracy: 0.6173 - val_loss: 0.6411 - val_accuracy: 0.6412
Epoch 9/10
377/377 [=====] - 18s 47ms/step - loss: 0.6599 - accuracy: 0.6164 - val_loss: 0.6358 - val_accuracy: 0.6362
Epoch 10/10
377/377 [=====] - 18s 48ms/step - loss: 0.6556 - accuracy: 0.6214 - val_loss: 0.6356 - val_accuracy: 0.6640
```

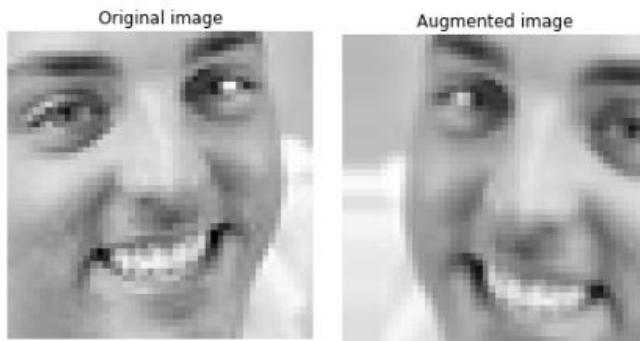
عرض تلخيص النموذج:



3. منحنى الضياع ومنحنى الدقة :(*loss & accuracy curves*)



في هذا النموذج قمنا بزيادة البيانات عن طريق (*data augmentation*)، على سبيل المثال:



النموذج السادس:

1. تقييم النموذج:

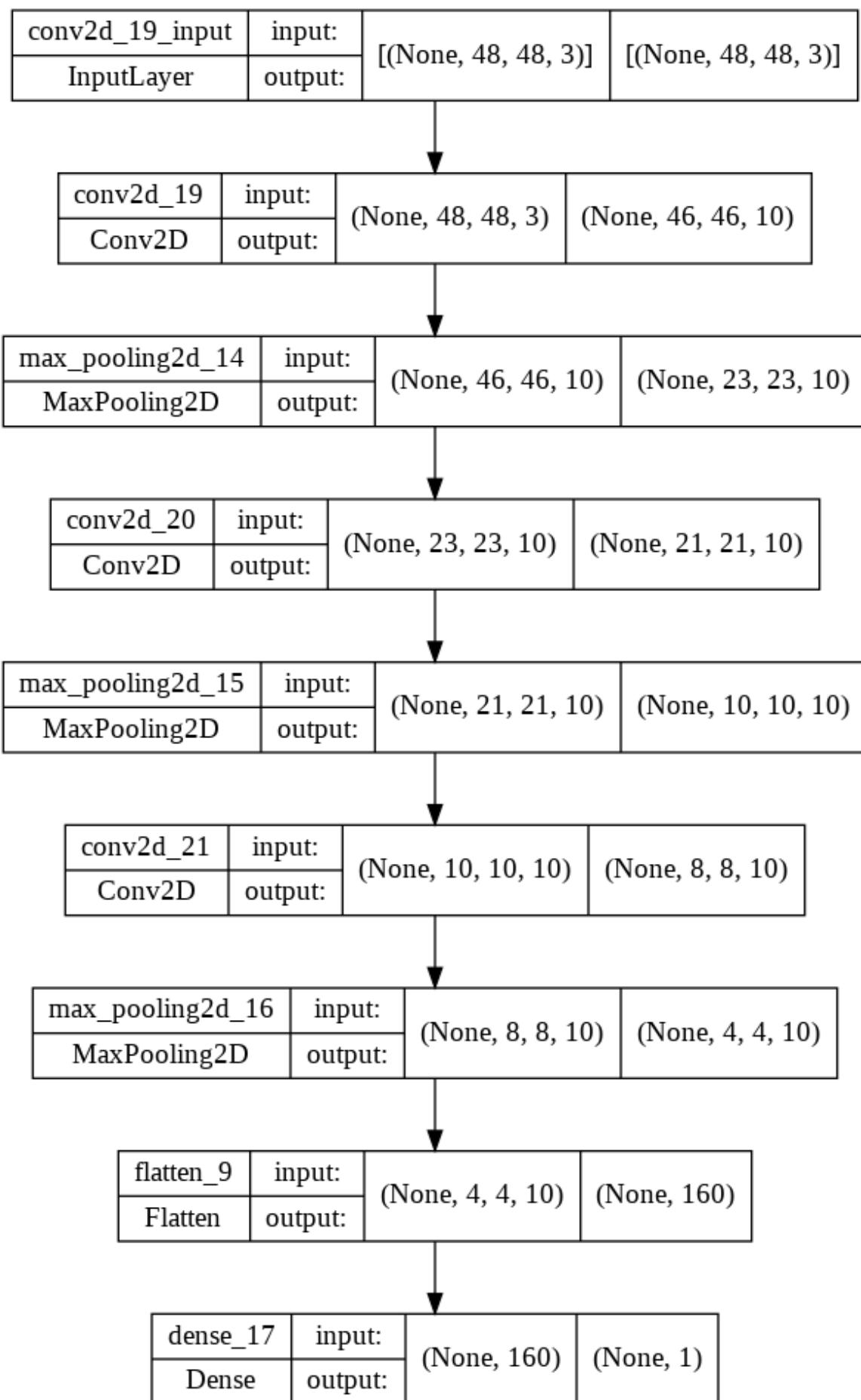
```
[ ] model_7.evaluate(train_data), model_7.evaluate(test_data)
```

```
377/377 [=====] - 5s 14ms/step - loss: 0.5369 - accuracy: 0.7460
95/95 [=====] - 1s 14ms/step - loss: 0.5512 - accuracy: 0.7276
([0.5369389653205872, 0.7460356950759888],
 [0.5511670708656311, 0.7275736331939697])
```

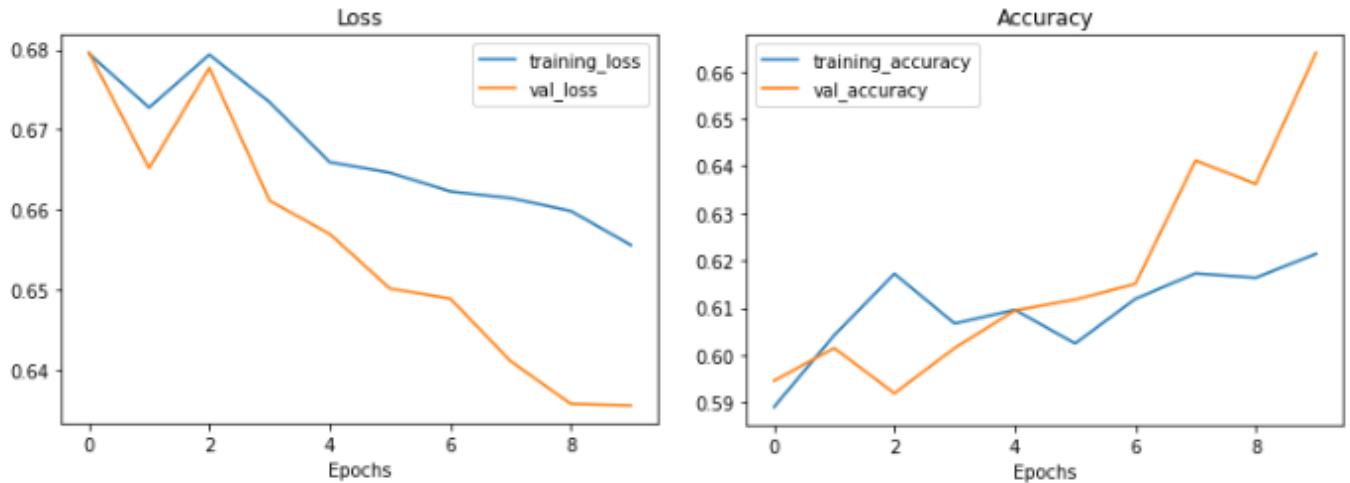
وقد تم تدريب النموذج لـ 10 دورات (*epochs*)

```
Epoch 1/10
377/377 [=====] - 23s 58ms/step - loss: 0.6689 - accuracy: 0.6008 - val_loss: 0.6672 - val_accuracy: 0.6124
Epoch 2/10
377/377 [=====] - 18s 48ms/step - loss: 0.6607 - accuracy: 0.6105 - val_loss: 0.6521 - val_accuracy: 0.6124
Epoch 3/10
377/377 [=====] - 18s 48ms/step - loss: 0.6569 - accuracy: 0.6164 - val_loss: 0.6533 - val_accuracy: 0.6147
Epoch 4/10
377/377 [=====] - 18s 47ms/step - loss: 0.6549 - accuracy: 0.6193 - val_loss: 0.6483 - val_accuracy: 0.6356
Epoch 5/10
377/377 [=====] - 18s 48ms/step - loss: 0.6489 - accuracy: 0.6276 - val_loss: 0.6378 - val_accuracy: 0.6488
Epoch 6/10
377/377 [=====] - 18s 48ms/step - loss: 0.6438 - accuracy: 0.6332 - val_loss: 0.6258 - val_accuracy: 0.6445
Epoch 7/10
377/377 [=====] - 18s 47ms/step - loss: 0.6340 - accuracy: 0.6425 - val_loss: 0.5963 - val_accuracy: 0.6895
Epoch 8/10
377/377 [=====] - 18s 47ms/step - loss: 0.6219 - accuracy: 0.6544 - val_loss: 0.6771 - val_accuracy: 0.6210
Epoch 9/10
377/377 [=====] - 18s 47ms/step - loss: 0.6198 - accuracy: 0.6530 - val_loss: 0.5711 - val_accuracy: 0.6951
Epoch 10/10
377/377 [=====] - 18s 47ms/step - loss: 0.6023 - accuracy: 0.6707 - val_loss: 0.5512 - val_accuracy: 0.7276
```

2. تلخيص النموذج:



3. منحنى الضياع ومنحنى الدقة (*loss & accuracy curves*)



في هذا النموذج قمنا بخلط مجموعات التدريب، من أجل أن يتعرف النموذج على كل نماذج البيانات خلال مرحلة التدريب، مما يجعله يتعلم ميزات عبر مصفوفة كبيرة من الصور.

انتهى الملف الأول

الملف الثاني

قمنا باعتماد مجموعة بيانات الصور كاملة.

المفاجأة	الحيادية	الحزن	السعادة	الخوف	الاشمئزاز	الغضب	
3171	4965	4830	7215	4097	436	3995	مجموعات التدريب
831	1233	1247	1774	1024	111	958	مجموعات الاختبار

النموذج الأول:

1. تقييم النموذج:

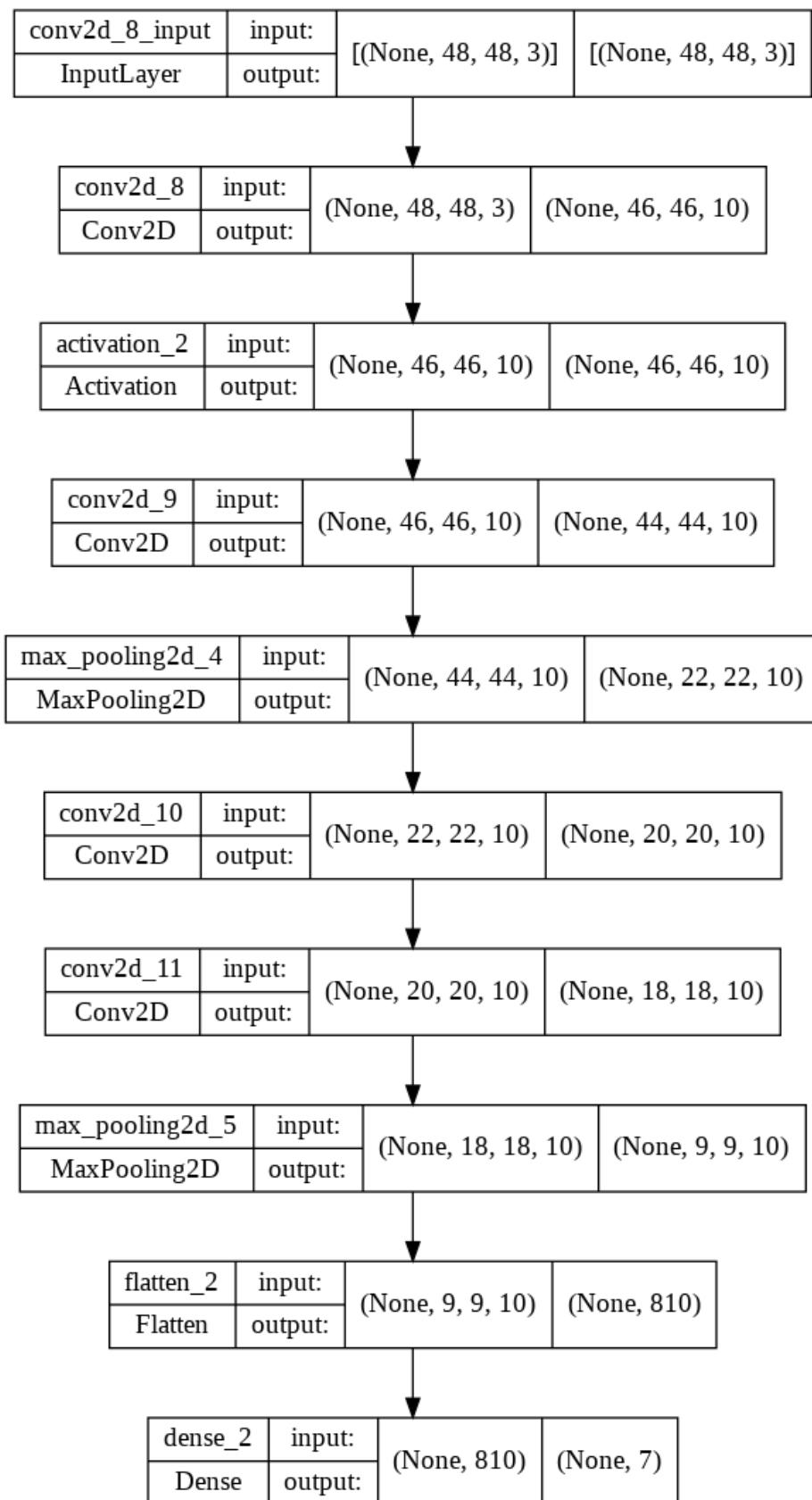
```
[ ] # Evaluate the loaded model
model_1.evaluate(train_data), model_1.evaluate(test_data)

898/898 [=====] - 14s 15ms/step - loss: 1.2682 - accuracy: 0.5238
225/225 [=====] - 3s 15ms/step - loss: 1.3468 - accuracy: 0.4819
([1.268236756324768, 0.5238426923751831],
 [1.3467594385147095, 0.48188909888267517])
```

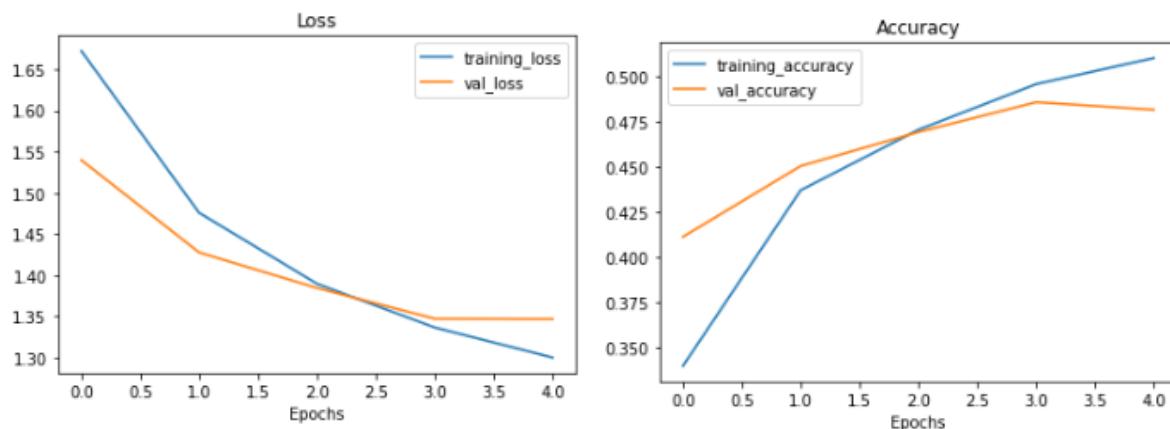
وقد تم تدريب النموذج لـ 5 دورات (epochs)

```
Epoch 1/5
898/898 [=====] - 19s 20ms/step - loss: 1.6722 - accuracy: 0.3398 - val_loss: 1.5397 - val_accuracy: 0.4114
Epoch 2/5
898/898 [=====] - 18s 20ms/step - loss: 1.4757 - accuracy: 0.4372 - val_loss: 1.4276 - val_accuracy: 0.4507
Epoch 3/5
898/898 [=====] - 17s 19ms/step - loss: 1.3896 - accuracy: 0.4708 - val_loss: 1.3844 - val_accuracy: 0.4695
Epoch 4/5
898/898 [=====] - 17s 19ms/step - loss: 1.3365 - accuracy: 0.4961 - val_loss: 1.3471 - val_accuracy: 0.4861
Epoch 5/5
898/898 [=====] - 17s 19ms/step - loss: 1.3000 - accuracy: 0.5106 - val_loss: 1.3468 - val_accuracy: 0.4819
```

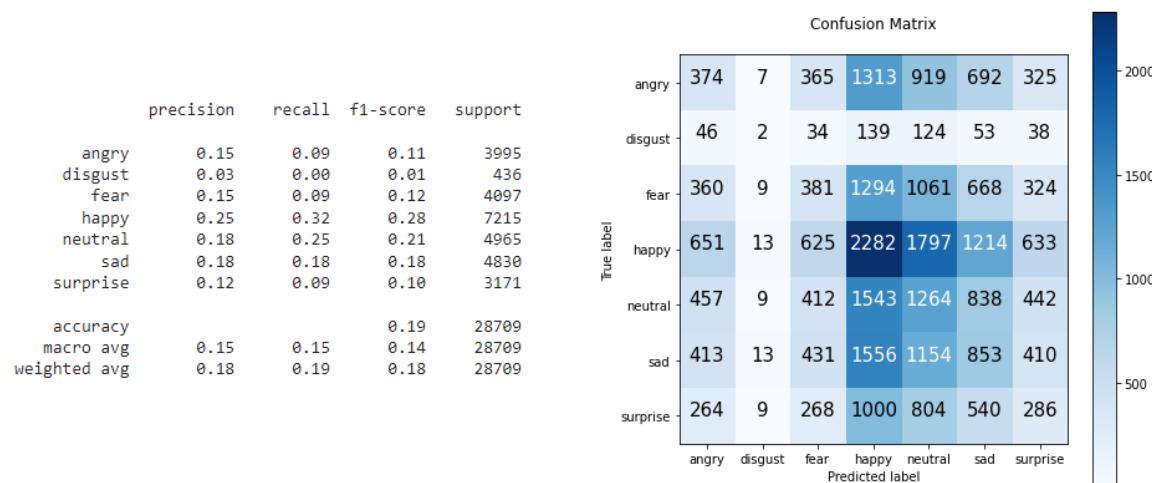
2. تلخيص النموذج:



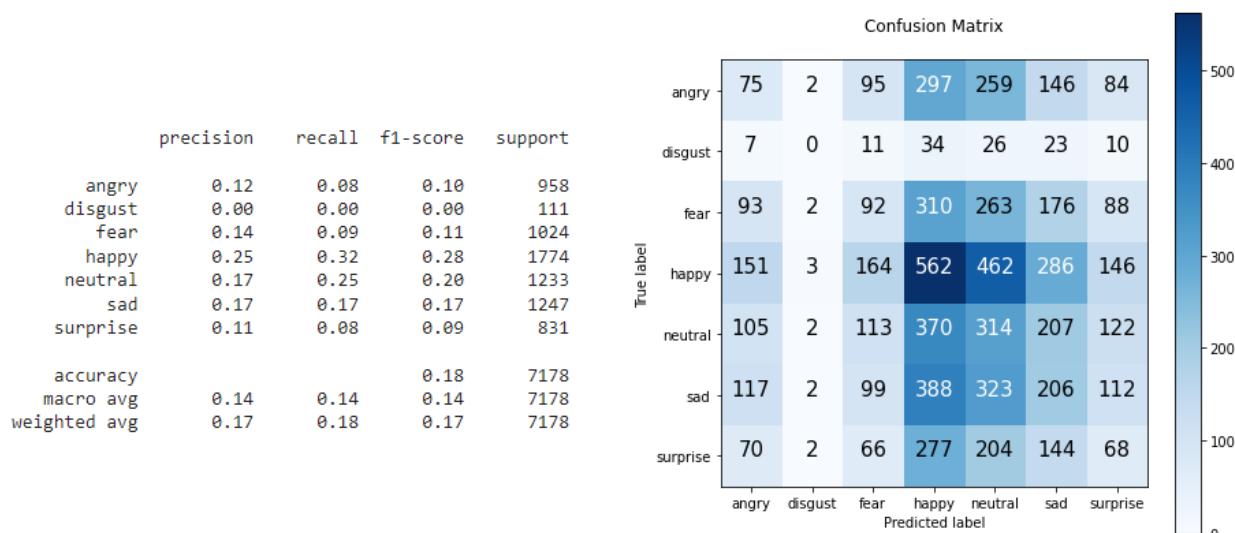
3. منحنى الضياع ومنحنى الدقة (loss & accuracy curves)



4. مصفوفة الاختلاط (confusion matrix) و报 告 (classification report) لمجموعات التدريب:



5. مصفوفة الاختلاط (confusion matrix) و报 告 (classification report) لمجموعات الاختبار:



النموذج الثاني:

1. تقييم النموذج:

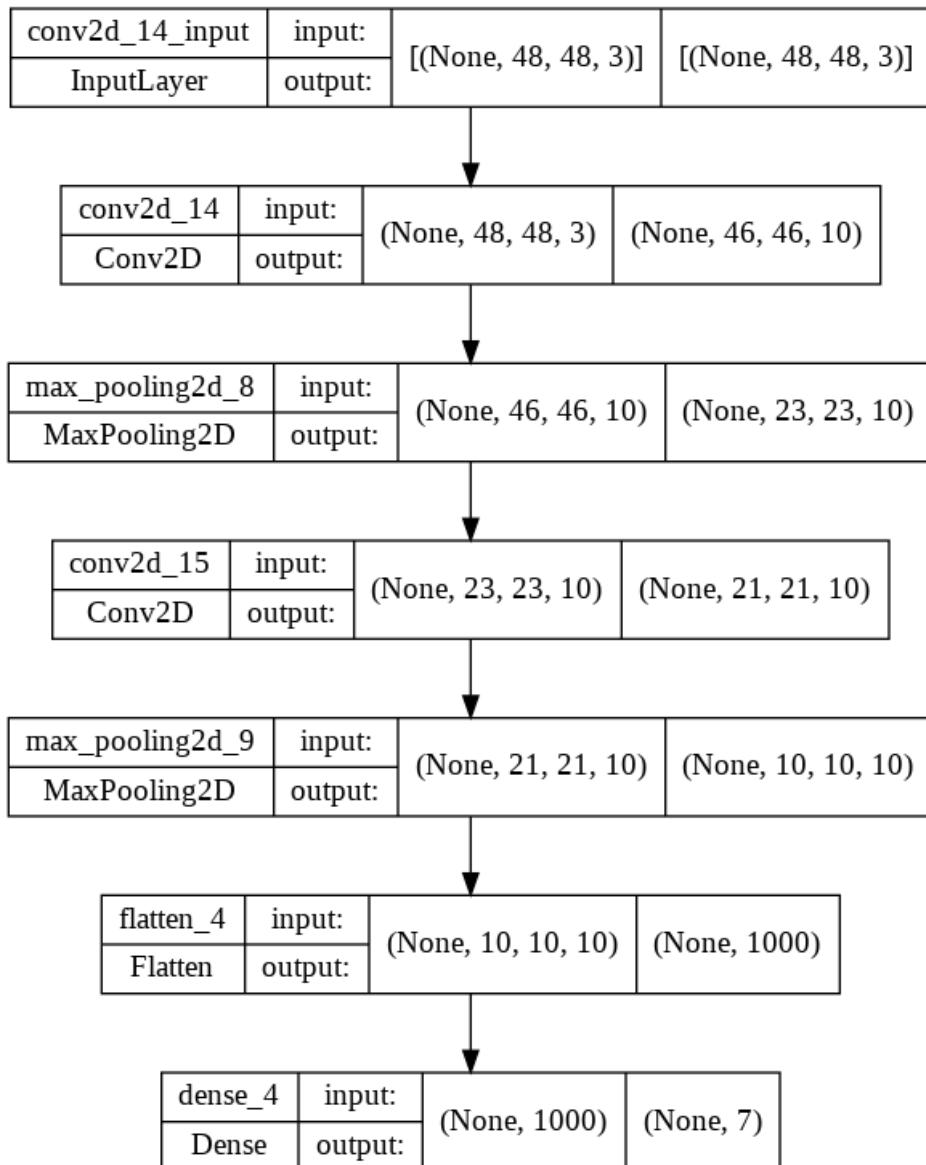
```
[ ] model_2.evaluate(train_data), model_2.evaluate(test_data)
```

```
898/898 [=====] - 12s 13ms/step - loss: 1.3290 - accuracy: 0.5049
225/225 [=====] - 3s 14ms/step - loss: 1.3934 - accuracy: 0.4724
([1.3290143013000488, 0.5049287676811218],
 [1.3934428691864014, 0.4724157154560089])
```

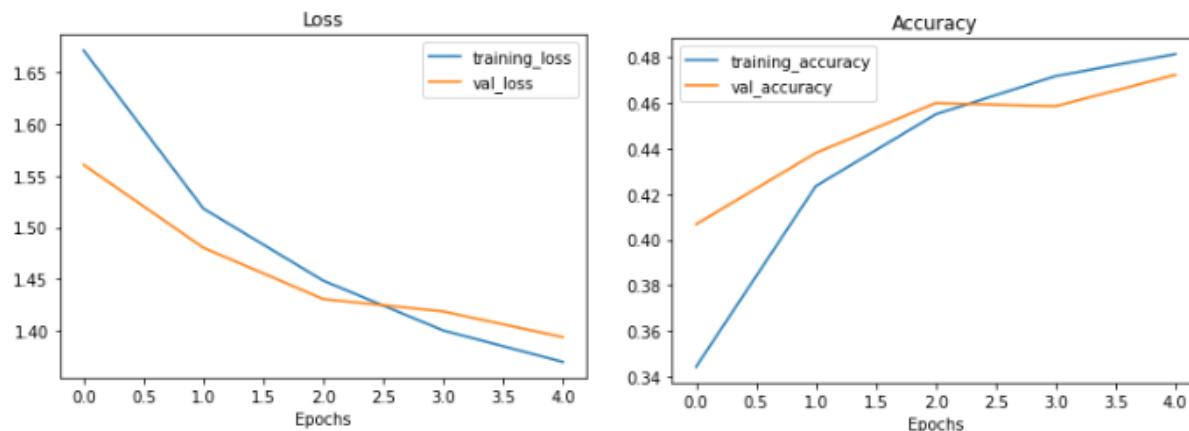
وقد تم تدريب النموذج لـ 5 دورات (epochs)

```
Epoch 1/5
898/898 [=====] - 17s 18ms/step - loss: 1.6717 - accuracy: 0.3442 - val_loss: 1.5606 - val_accuracy: 0.4068
Epoch 2/5
898/898 [=====] - 16s 18ms/step - loss: 1.5182 - accuracy: 0.4236 - val_loss: 1.4802 - val_accuracy: 0.4381
Epoch 3/5
898/898 [=====] - 17s 18ms/step - loss: 1.4482 - accuracy: 0.4551 - val_loss: 1.4304 - val_accuracy: 0.4600
Epoch 4/5
898/898 [=====] - 17s 19ms/step - loss: 1.4000 - accuracy: 0.4718 - val_loss: 1.4184 - val_accuracy: 0.4586
Epoch 5/5
898/898 [=====] - 17s 19ms/step - loss: 1.3694 - accuracy: 0.4815 - val_loss: 1.3934 - val_accuracy: 0.4724
```

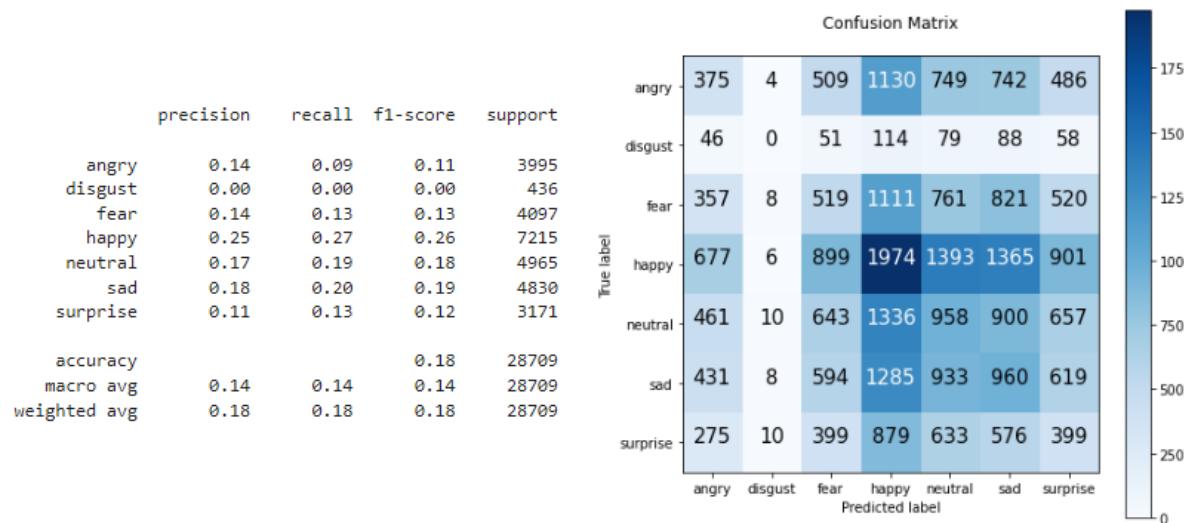
2. تلخيص النموذج:



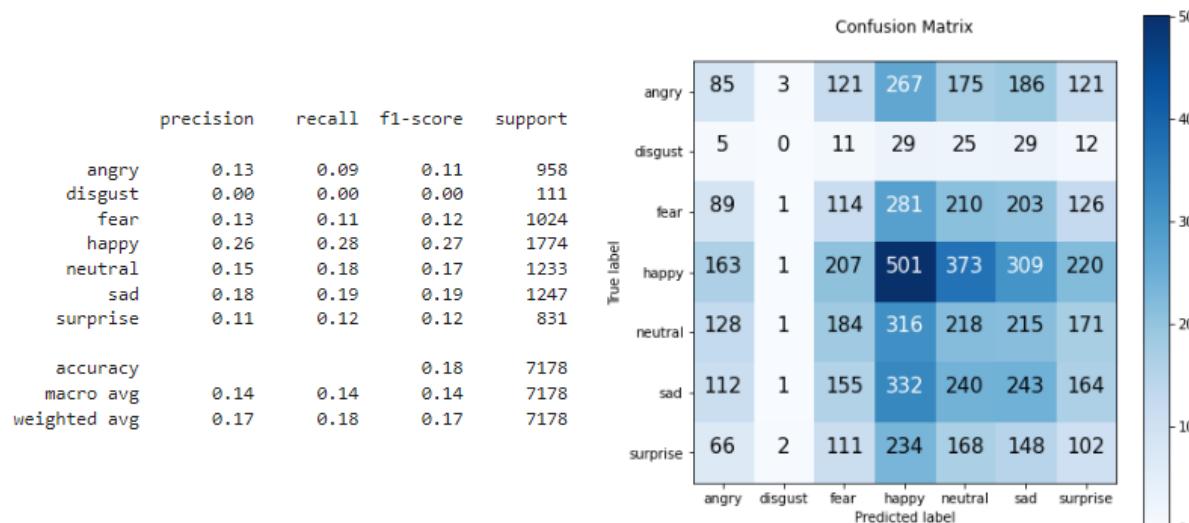
3. منحنى الخسارة ومنحنى الدقة (loss & accuracy curves)



4. مصفوفة الاختلاط (confusion matrix) و报 告 (classification report) لمجموعات التدريب:



5. مصفوفة الاختلاط (confusion matrix) و报 告 (classification report) لمجموعات الاختبار:



النموذج الثالث:

1. تقييم النموذج:

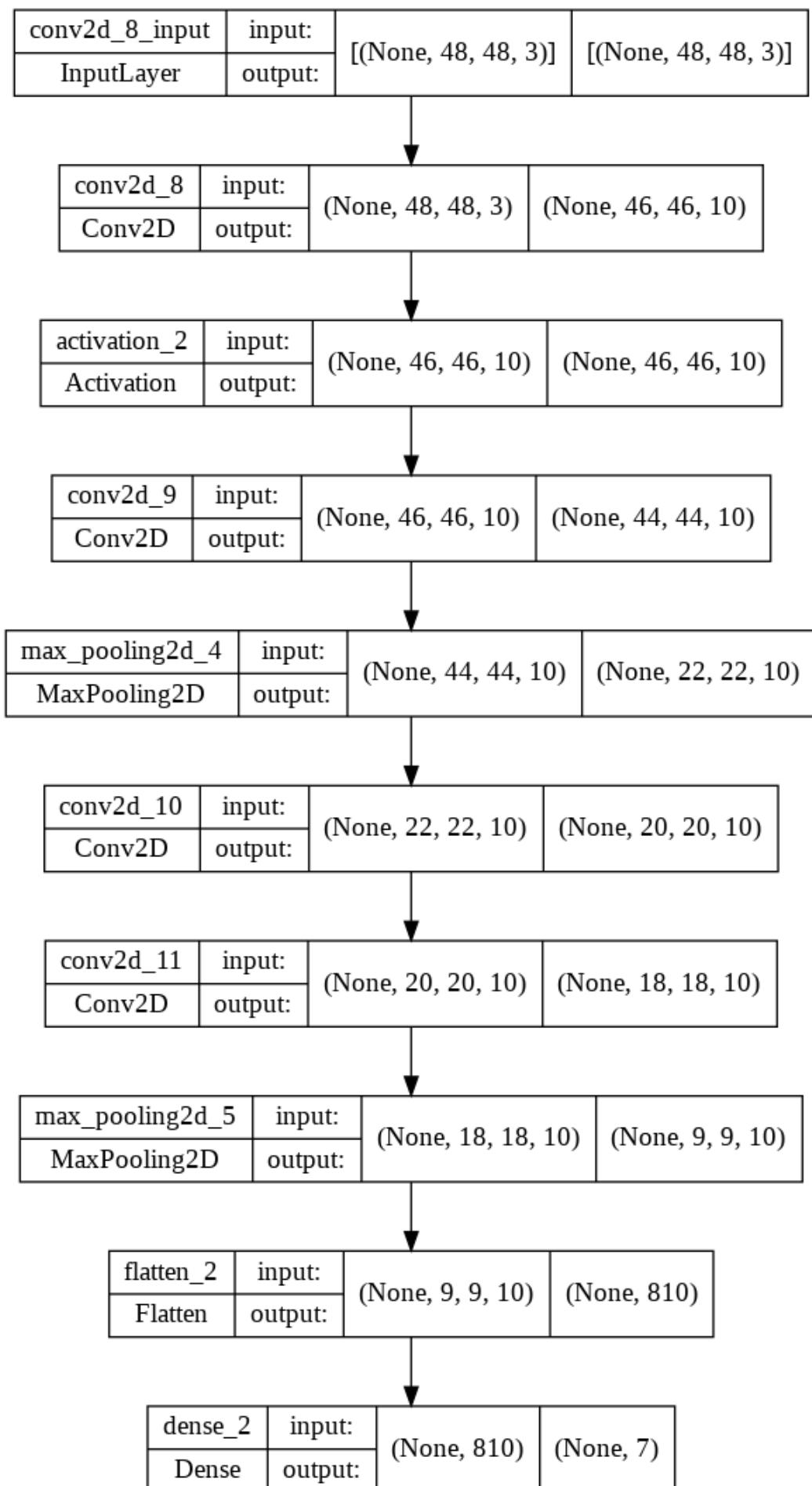
```
[ ] model_3.evaluate(train_data), model_3.evaluate(test_data)

898/898 [=====] - 13s 14ms/step - loss: 1.5684 - accuracy: 0.3826
225/225 [=====] - 3s 14ms/step - loss: 1.5672 - accuracy: 0.3799
([1.5683972835540771, 0.38259780406951904],
 [1.5671793222427368, 0.37991082668304443])
```

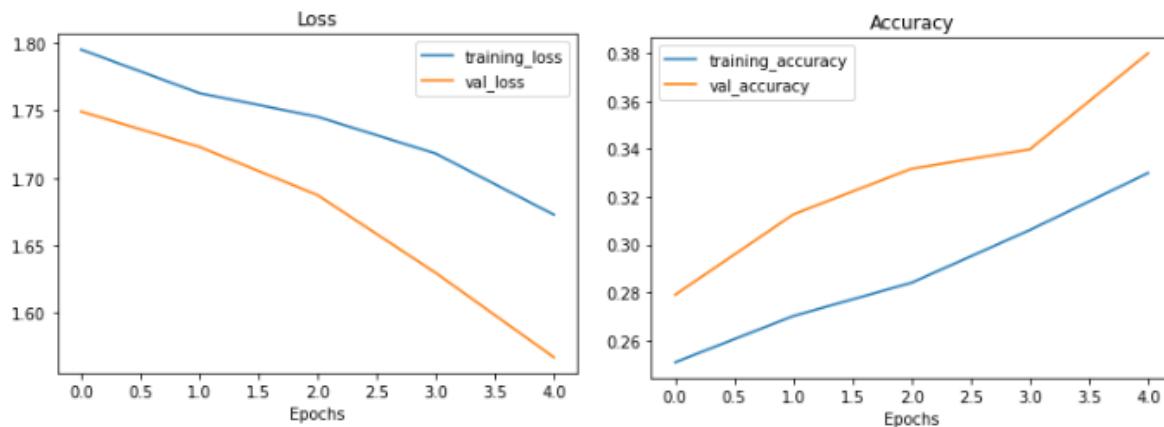
وقد تم تدريب النموذج لـ 5 دورات (*epochs*)

```
Epoch 1/5
898/898 [=====] - 44s 48ms/step - loss: 1.7951 - accuracy: 0.2509 - val_loss: 1.7492 - val_accuracy: 0.2790
Epoch 2/5
898/898 [=====] - 42s 47ms/step - loss: 1.7629 - accuracy: 0.2702 - val_loss: 1.7231 - val_accuracy: 0.3126
Epoch 3/5
898/898 [=====] - 43s 48ms/step - loss: 1.7453 - accuracy: 0.2841 - val_loss: 1.6874 - val_accuracy: 0.3317
Epoch 4/5
898/898 [=====] - 43s 47ms/step - loss: 1.7181 - accuracy: 0.3061 - val_loss: 1.6297 - val_accuracy: 0.3398
Epoch 5/5
898/898 [=====] - 46s 52ms/step - loss: 1.6728 - accuracy: 0.3299 - val_loss: 1.5672 - val_accuracy: 0.3799
```

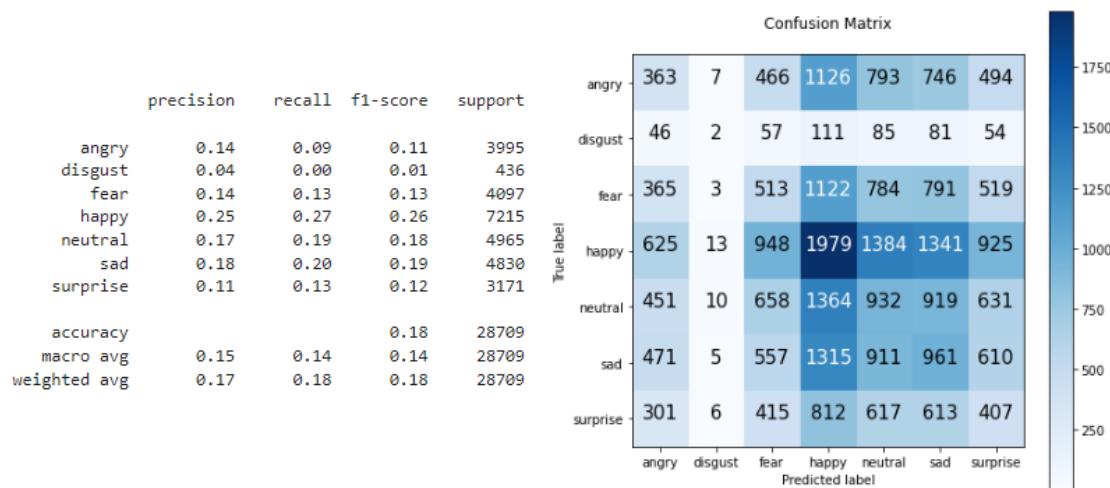
2. تلخيص النموذج:



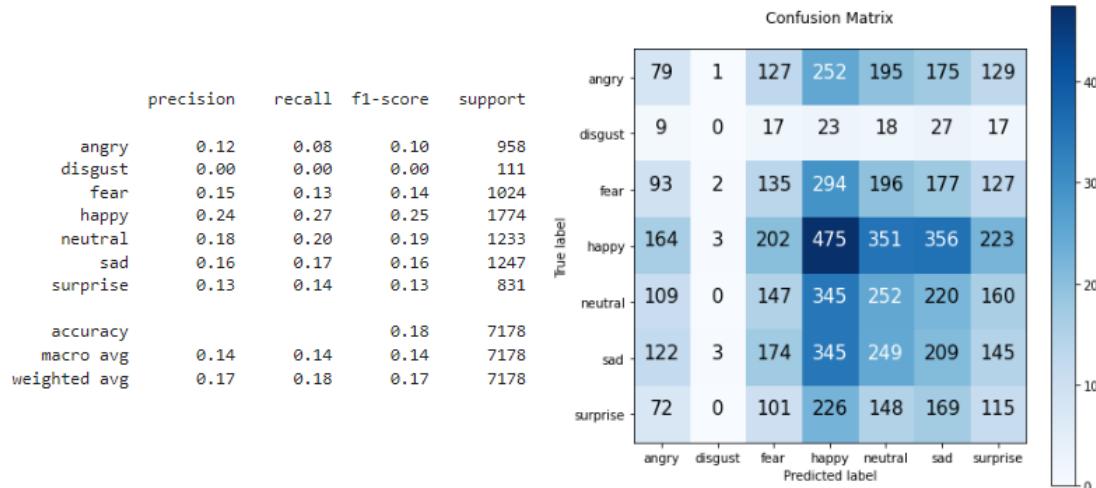
3. منحنى الخسارة ومنحنى الدقة (loss & accuracy curves)



4. مصفوفة الاختلاط (confusion matrix) وتقرير التصنيف (classification report) لمجموعات التدريب:



5. مصفوفة الاختلاط (confusion matrix) وتقرير التصنيف (classification report) لمجموعات الاختبار:



انتهى الملف الثاني

الملف الثالث

قمنا باعتماد جزء من مجموعة بيانات الصور الكاملة كالتالي:

المفاجأة	الحيادية	الحزن	السعادة	الخوف	الاشمئاز	الغضب	مجموعات التدريب
100	100	100	100	100	100	100	مجموعات التدريب
مجموعات الاختبار							
250	250	250	250	250	250	250	

قمنا باستعمال جزء بسيط من مجموعة البيانات الكاملة، والسبب هو أن نرى إذا كنا نستطيع الحصول على دقة بمجموعة بيانات بسيطة بالاعتماد على شبكة مدربة سابقاً على ملايين الصور.

النموذج الأول: قمنا ببناء نموذجنا على شبكة (*Resnet50*), وقمنا باستخراج السمات وقمنا بأخذ أوزان هذه الشبكة وضبط مخرجاتها لتكون ملائمة لمشكلتنا.

تقييم النموذج:

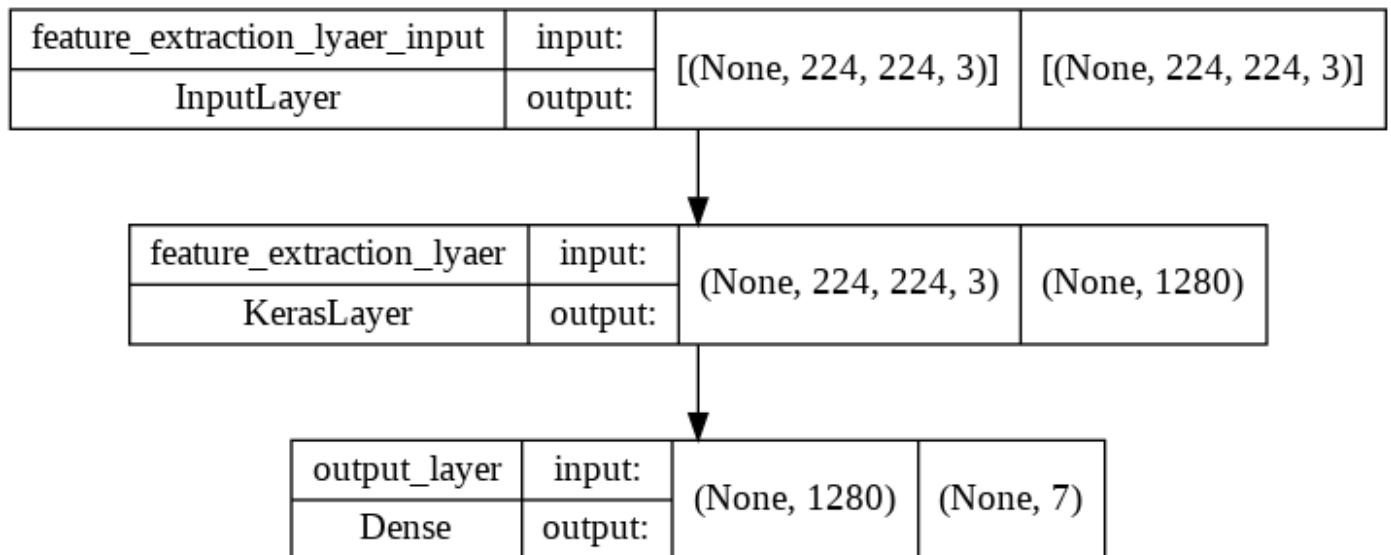
```
[ ] # Evaluate the loaded model
resnet_model.evaluate(train_mini_data), resnet_model.evaluate(test_data)

22/22 [=====] - 2s 84ms/step - loss: 0.8280 - accuracy: 0.7829
55/55 [=====] - 5s 84ms/step - loss: 1.7018 - accuracy: 0.3749
([0.8279626369476318, 0.7828571200370789],
 [1.7017521858215332, 0.3748571574687958])
```

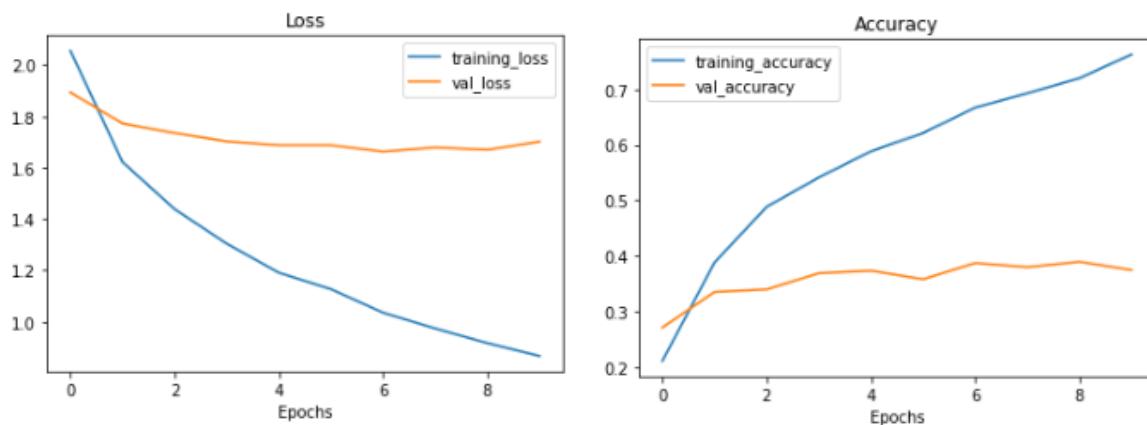
وقد تم تدريب النموذج لـ **10** دورات (*epochs*) :

```
Epoch 1/10
22/22 [=====] - 23s 404ms/step - loss: 2.0556 - accuracy: 0.2114 - val_loss: 1.8937 - val_accuracy: 0.2714
Epoch 2/10
22/22 [=====] - 6s 293ms/step - loss: 1.6220 - accuracy: 0.3886 - val_loss: 1.7724 - val_accuracy: 0.3354
Epoch 3/10
22/22 [=====] - 6s 294ms/step - loss: 1.4386 - accuracy: 0.4886 - val_loss: 1.7351 - val_accuracy: 0.3400
Epoch 4/10
22/22 [=====] - 6s 294ms/step - loss: 1.3040 - accuracy: 0.5414 - val_loss: 1.7026 - val_accuracy: 0.3691
Epoch 5/10
22/22 [=====] - 6s 296ms/step - loss: 1.1910 - accuracy: 0.5886 - val_loss: 1.6879 - val_accuracy: 0.3737
Epoch 6/10
22/22 [=====] - 7s 329ms/step - loss: 1.1268 - accuracy: 0.6214 - val_loss: 1.6879 - val_accuracy: 0.3583
Epoch 7/10
22/22 [=====] - 6s 297ms/step - loss: 1.0352 - accuracy: 0.6671 - val_loss: 1.6624 - val_accuracy: 0.3874
Epoch 8/10
22/22 [=====] - 6s 302ms/step - loss: 0.9737 - accuracy: 0.6929 - val_loss: 1.6791 - val_accuracy: 0.3800
Epoch 9/10
22/22 [=====] - 6s 301ms/step - loss: 0.9167 - accuracy: 0.7200 - val_loss: 1.6698 - val_accuracy: 0.3897
Epoch 10/10
22/22 [=====] - 6s 300ms/step - loss: 0.8666 - accuracy: 0.7629 - val_loss: 1.7018 - val_accuracy: 0.3749
```

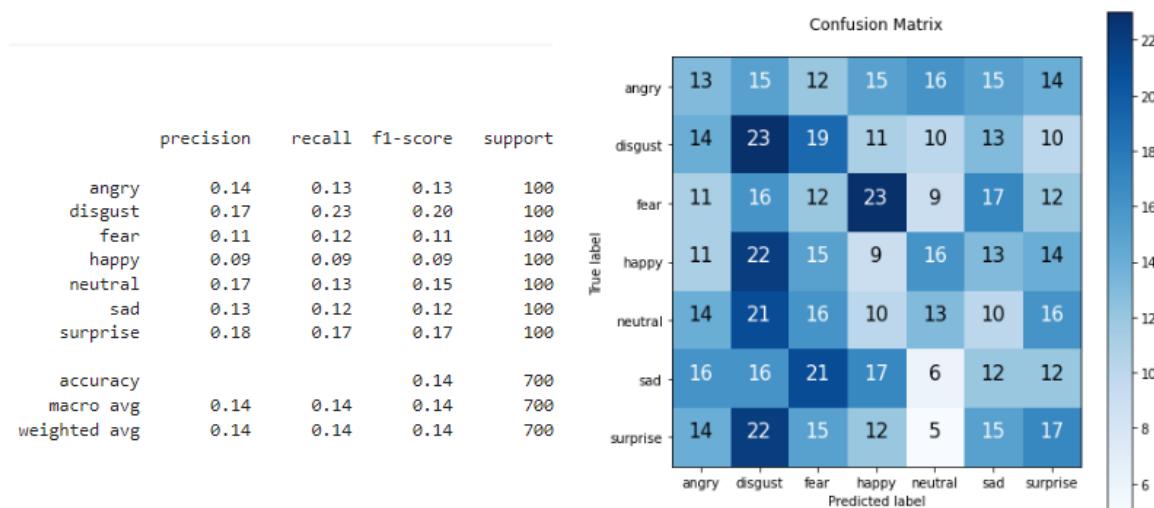
2. تلخيص النموذج:



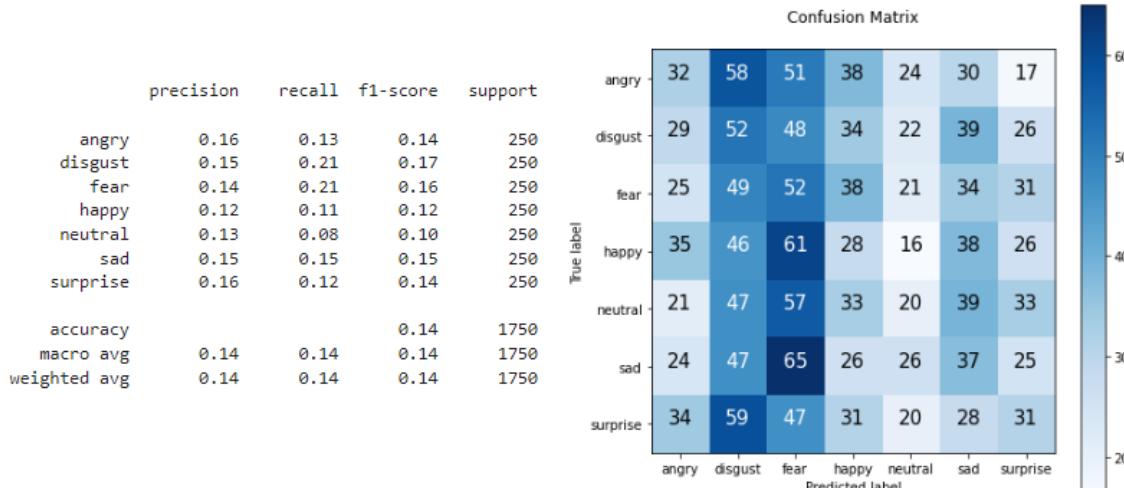
3. منحنى الخسارة ومنحنى الدقة (loss & accuracy curves)



4. مصفوفة الاختلاط (confusion matrix) وتقرير التصنيف (classification report) لمجموعات التدريب:



5. مصفوفة الاختلاط (confusion matrix) وتقرير التصنيف (classification report) لمجموعات الاختبار:



النموذج الثاني: قمنا ببناء نموذجنا على شبكة (EfficientNetB0)، وقمنا باستخراج السمات وقمنا بأخذ أوزان هذه الشبكة وضبط مخرجاتها لتكون ملائمة لمشكلتنا.

1. تقييم النموذج:

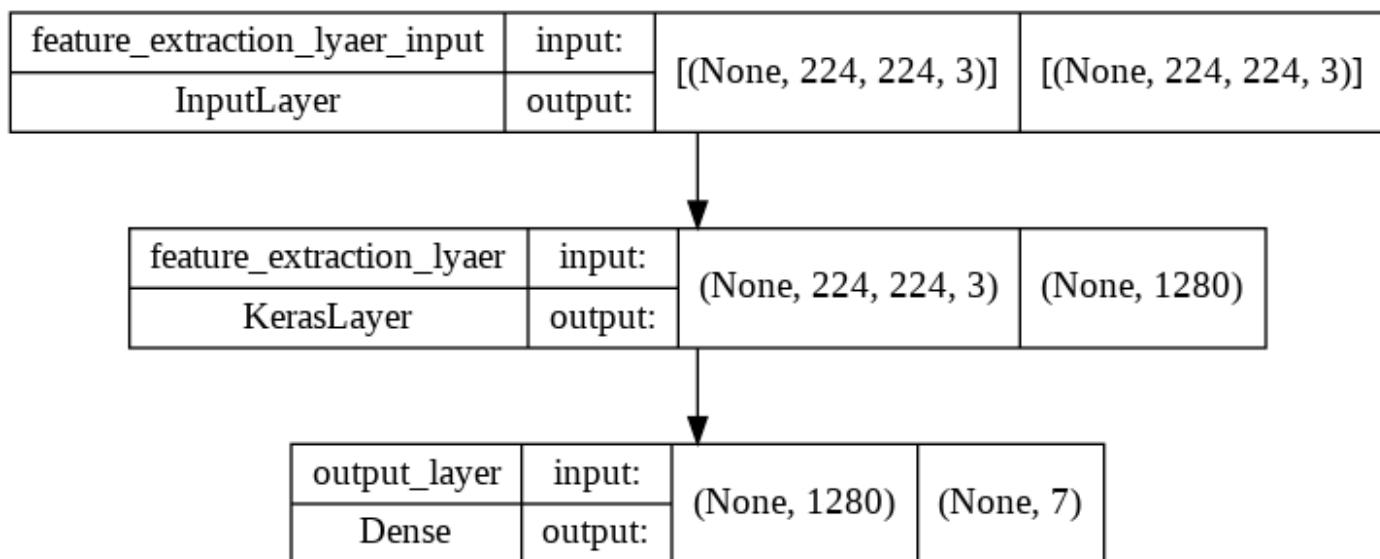
```
[ ] # Evaluate the loaded model
efficientnet_model.evaluate(train_mini_data), efficientnet_model.evaluate(test_data)

22/22 [=====] - 1s 63ms/step - loss: 1.3583 - accuracy: 0.5414
55/55 [=====] - 4s 63ms/step - loss: 1.6937 - accuracy: 0.3440
([1.3582607507705688, 0.5414285659790039],
 [1.6937249898910522, 0.3440000116825104])
```

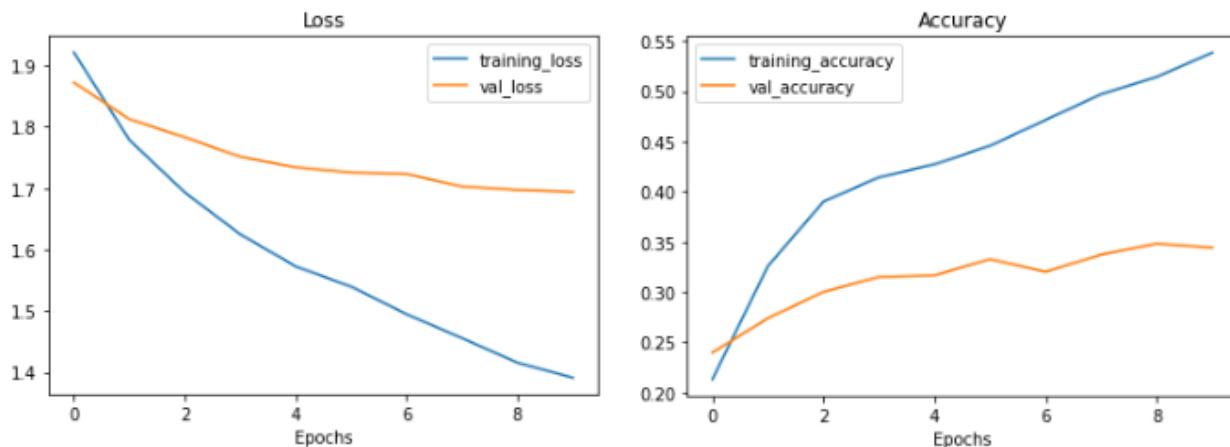
وقد تم تدريب النموذج لـ 10 دورات (epochs)

```
Epoch 1/10
22/22 [=====] - 15s 303ms/step - loss: 1.9214 - accuracy: 0.2129 - val_loss: 1.8724 - val_accuracy: 0.2394
Epoch 2/10
22/22 [=====] - 5s 219ms/step - loss: 1.7787 - accuracy: 0.3257 - val_loss: 1.8125 - val_accuracy: 0.2737
Epoch 3/10
22/22 [=====] - 5s 218ms/step - loss: 1.6931 - accuracy: 0.3900 - val_loss: 1.7831 - val_accuracy: 0.3000
Epoch 4/10
22/22 [=====] - 5s 217ms/step - loss: 1.6248 - accuracy: 0.4143 - val_loss: 1.7512 - val_accuracy: 0.3149
Epoch 5/10
22/22 [=====] - 5s 218ms/step - loss: 1.5725 - accuracy: 0.4271 - val_loss: 1.7342 - val_accuracy: 0.3166
Epoch 6/10
22/22 [=====] - 5s 219ms/step - loss: 1.5395 - accuracy: 0.4457 - val_loss: 1.7253 - val_accuracy: 0.3326
Epoch 7/10
22/22 [=====] - 5s 217ms/step - loss: 1.4945 - accuracy: 0.4714 - val_loss: 1.7230 - val_accuracy: 0.3200
Epoch 8/10
22/22 [=====] - 5s 218ms/step - loss: 1.4555 - accuracy: 0.4971 - val_loss: 1.7027 - val_accuracy: 0.3371
Epoch 9/10
22/22 [=====] - 5s 218ms/step - loss: 1.4154 - accuracy: 0.5143 - val_loss: 1.6970 - val_accuracy: 0.3480
Epoch 10/10
22/22 [=====] - 5s 220ms/step - loss: 1.3909 - accuracy: 0.5386 - val_loss: 1.6937 - val_accuracy: 0.3440
```

2. تلخيص النموذج:

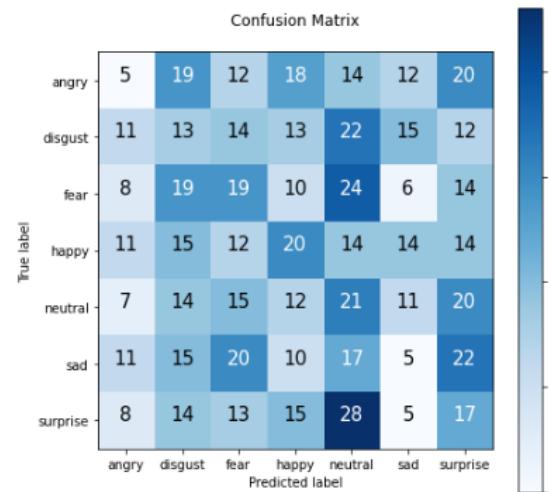


3. منحنى الخسارة ومنحنى الدقة (loss & accuracy curves)



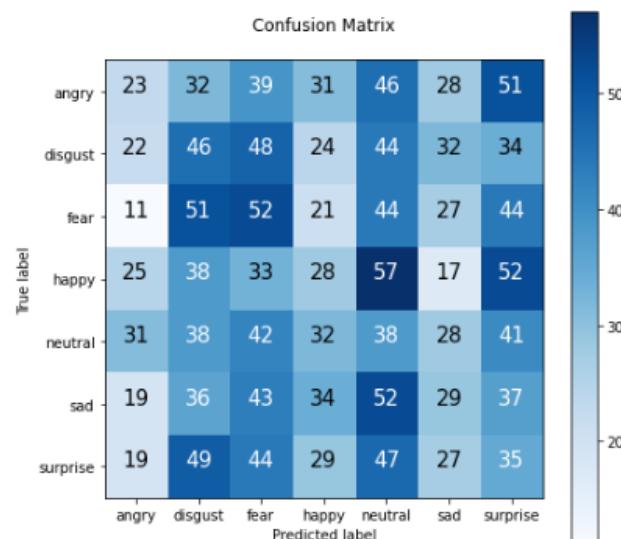
4. مصفوفة الاختلاط (confusion matrix) وتقرير التصنيف (classification report) لمجموعات التدريب:

	precision	recall	f1-score	support
angry	0.08	0.05	0.06	100
disgust	0.12	0.13	0.12	100
fear	0.18	0.19	0.19	100
happy	0.20	0.20	0.20	100
neutral	0.15	0.21	0.18	100
sad	0.07	0.05	0.06	100
surprise	0.14	0.17	0.16	100
accuracy			0.14	700
macro avg	0.14	0.14	0.14	700
weighted avg	0.14	0.14	0.14	700



5. مصفوفة الاختلاط (confusion matrix) وتقرير التصنيف (classification report) لمجموعات الاختبار:

	precision	recall	f1-score	support
angry	0.15	0.09	0.11	250
disgust	0.16	0.18	0.17	250
fear	0.17	0.21	0.19	250
happy	0.14	0.11	0.12	250
neutral	0.12	0.15	0.13	250
sad	0.15	0.12	0.13	250
surprise	0.12	0.14	0.13	250
accuracy			0.14	1750
macro avg	0.14	0.14	0.14	1750
weighted avg	0.14	0.14	0.14	1750



انتهى الملف الثالث

الملف الرابع

في البداية، قمنا باعتماد جزء من مجموعة بيانات الصور الكاملة، كالتالي:

المفاجأة	الحيادية	الحزن	السعادة	الخوف	الاشمئاز	الغضب	
100	100	100	100	100	100	100	مجموعات التدريب
250	250	250	250	250	250	250	مجموعات الاختبار

النموذج الابتدائي: قمنا ببناء نموذجنا على شبكة (*EfficientNetB0*), وقمنا باستخراج السمات وقمنا بأخذ أوزان هذه الشبكة وضبط مخرجاتها لتكون ملائمة لمشكلتنا.

1. تقييم النموذج:

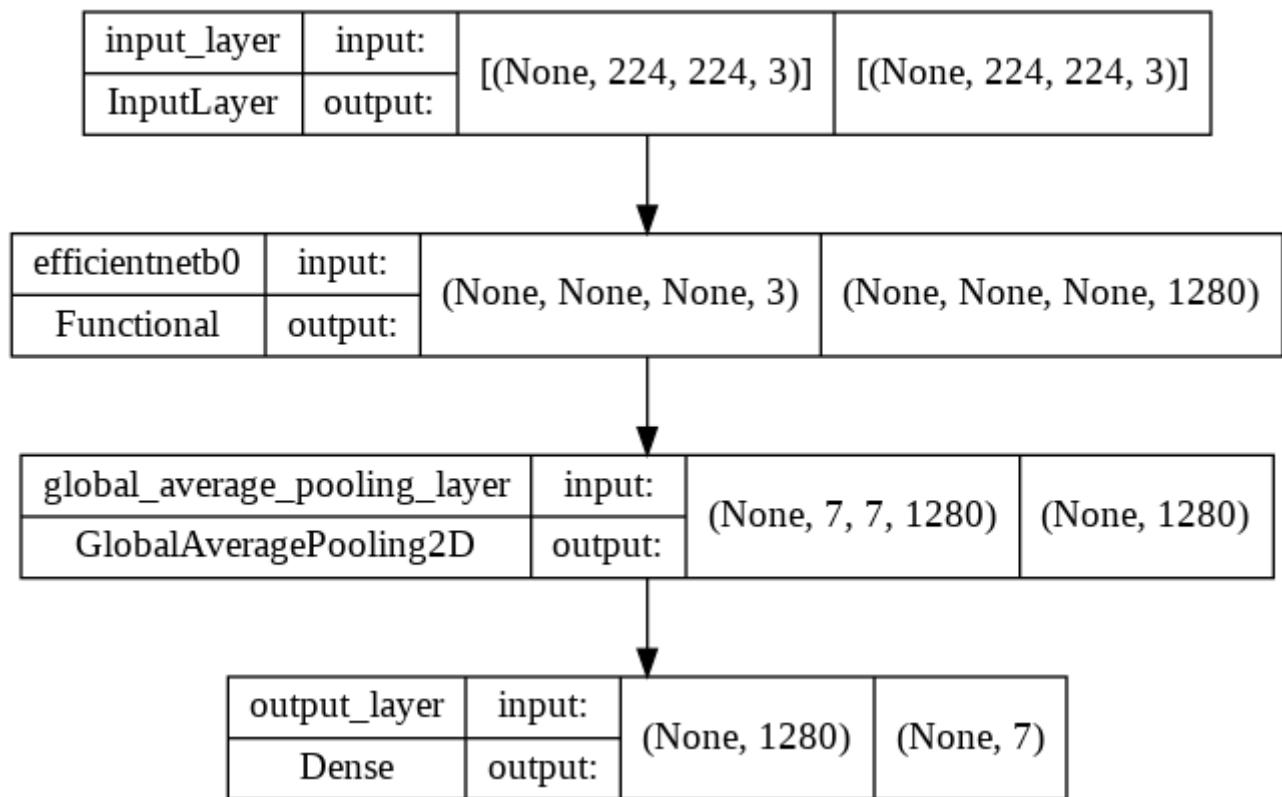
```
[ ] # Evaluate on the full test dataset
model_0.evaluate(test_data)

55/55 [=====] - 8s 142ms/step - loss: 1.7218 - accuracy: 0.3297
[1.7218068838119507, 0.329714298248291]
```

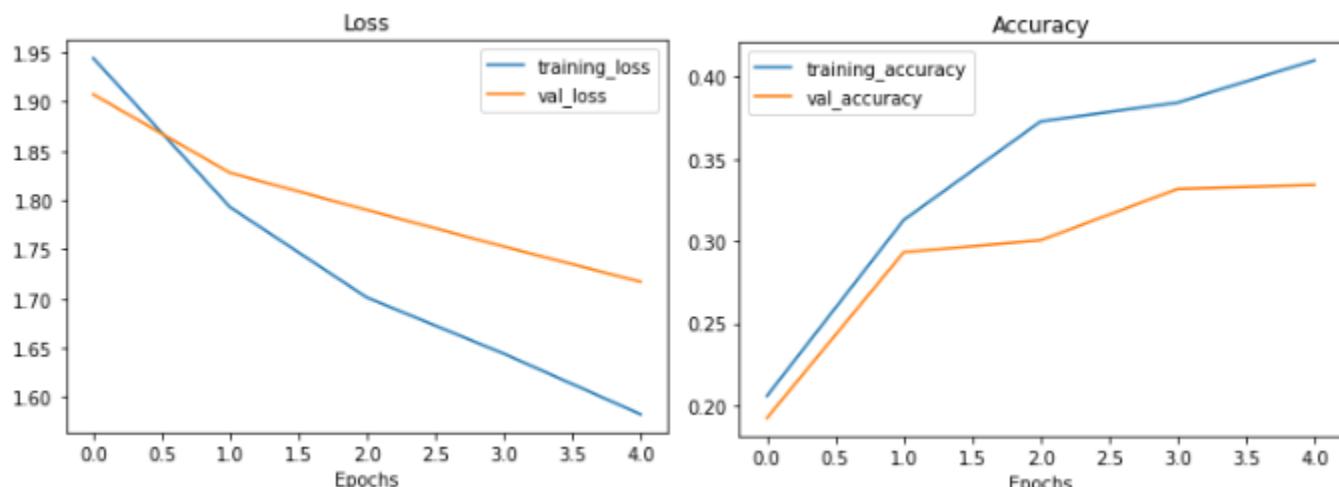
وقد تم تدريب النموذج لـ 5 دورات (*epochs*):

```
Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb0_notop.h5
16711680/16705208 [=====] - 0s 0us/step
16719872/16705208 [=====] - 0s 0us/step
Shape after passing inputs through base model: (None, 7, 7, 1280)
Shape after GlobalAveragePooling2D: (None, 1280)
Saving TensorBoard log files to: transfer_learning/mini_data_feature_extraction/20220405-194602
Epoch 1/5
22/22 [=====] - 24s 340ms/step - loss: 1.9443 - accuracy: 0.2057 - val_loss: 1.9076 - val_accuracy: 0.1923
Epoch 2/5
22/22 [=====] - 5s 226ms/step - loss: 1.7930 - accuracy: 0.3129 - val_loss: 1.8278 - val_accuracy: 0.2933
Epoch 3/5
22/22 [=====] - 6s 256ms/step - loss: 1.7012 - accuracy: 0.3729 - val_loss: 1.7901 - val_accuracy: 0.3005
Epoch 4/5
22/22 [=====] - 5s 228ms/step - loss: 1.6442 - accuracy: 0.3843 - val_loss: 1.7529 - val_accuracy: 0.3317
Epoch 5/5
22/22 [=====] - 5s 224ms/step - loss: 1.5826 - accuracy: 0.4100 - val_loss: 1.7172 - val_accuracy: 0.3341
```

2. تلخيص النموذج:



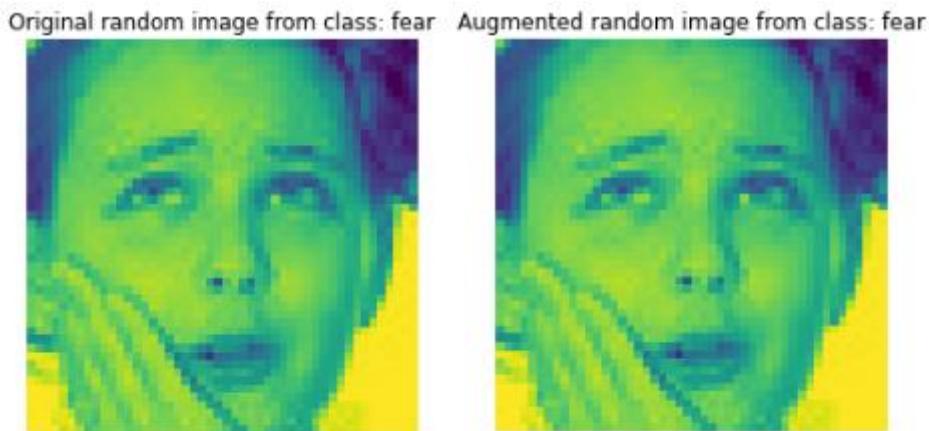
3. منحني الخسارة ومنحني الدقة (loss & accuracy curves)



وبعدها قمنا بتنقیل مجموعة صور البيانات لتصبح كالتالي:

الملائجة	الحيانية	الحزن	السعادة	الخوف	الاشمئاز	الغصب	
10	10	10	10	10	10	10	مجموعات التدريب
250	250	250	250	250	250	250	مجموعات الاختبار

النموذج الأول: قمنا ببناء نموذجنا على شبكة (*EfficientNetB0*)، وقمنا باستخراج السمات وقمنا بأخذ أوزان هذه الشبكة وضبط مخرجاتها لتكون ملائمة لمشكلتنا، وأيضاً قمنا بزيادة البيانات عن طريق **على سبيل المثال: (*data augmentation*)**



1. تقييم النموذج:

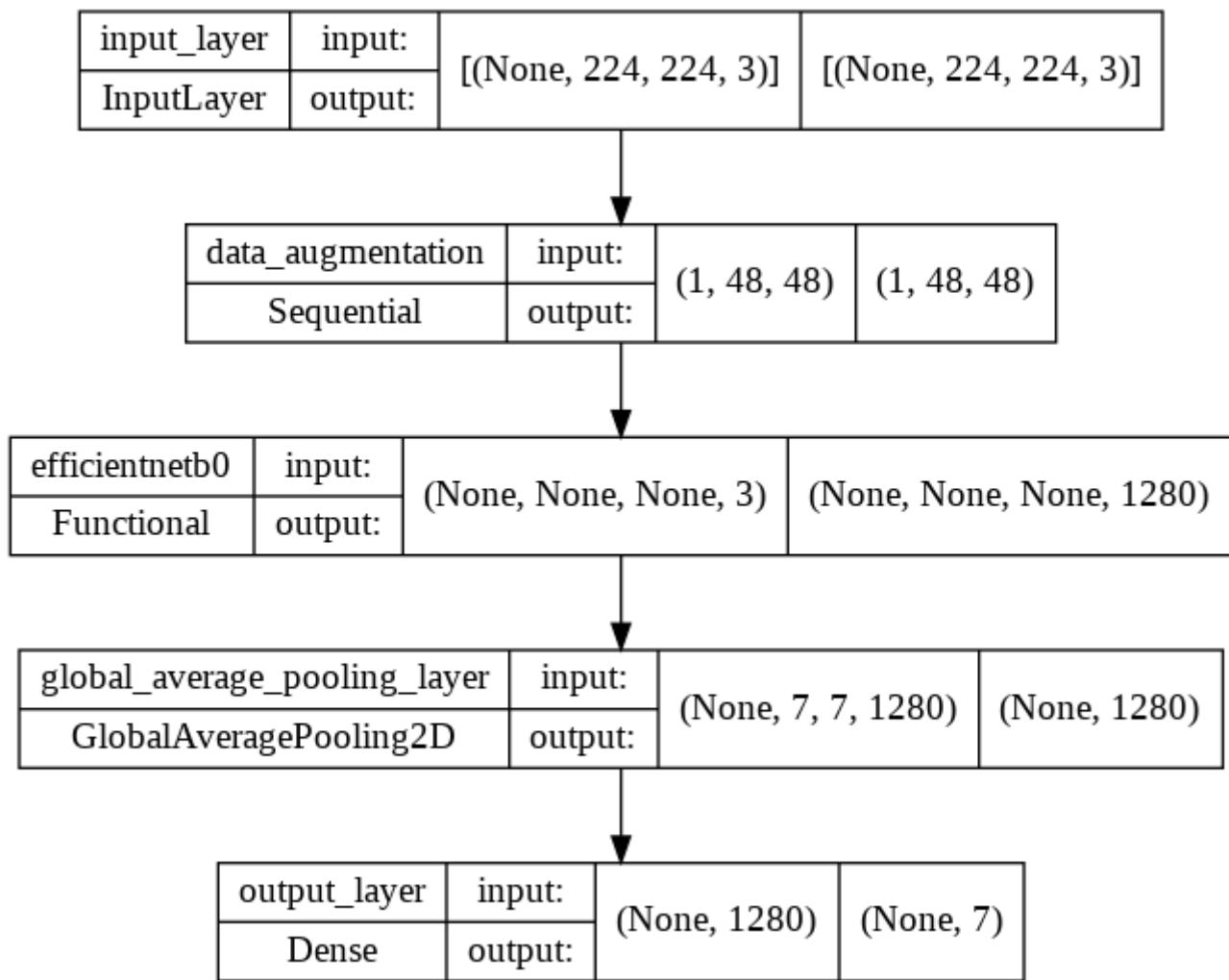
```
[ ] # Evaluate on the full test dataset
results_1_percent_data_aug = model_1.evaluate(test_data)
results_1_percent_data_aug

55/55 [=====] - 8s 132ms/step - loss: 1.9592 - accuracy: 0.1834
[1.959230899810791, 0.1834285706281662]
```

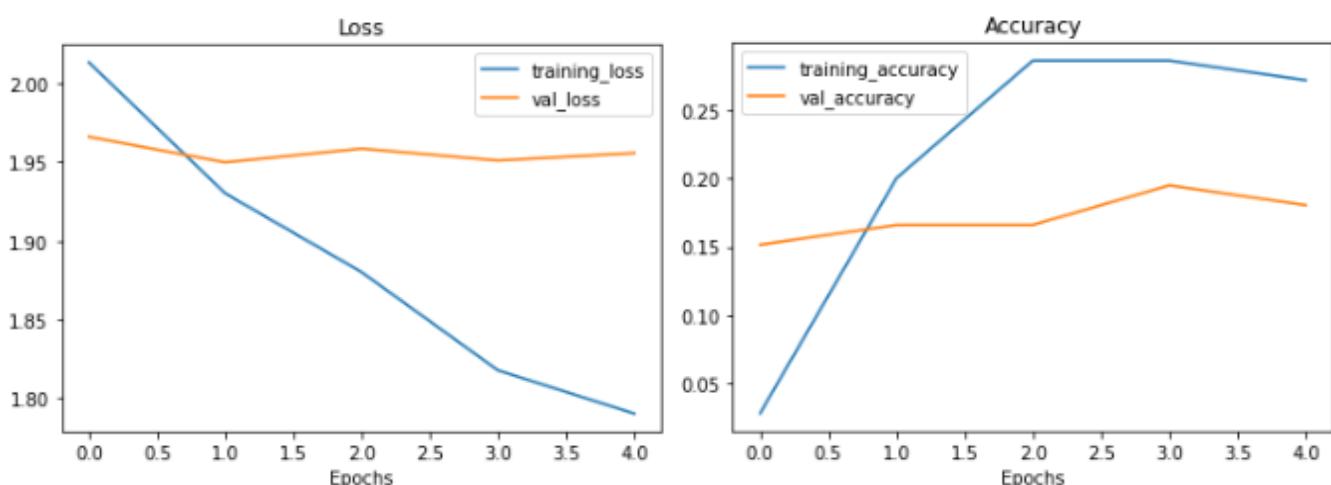
وقد تم تدريب النموذج لـ 5 دورات (*epochs*):

```
3/3 [=====] - ETA: 0s - loss: 2.0135 - accuracy: 0.0286
3/3 [=====] - 14s 2s/step - loss: 2.0135 - accuracy: 0.0286 - val_loss: 1.9662 - val_accuracy: 0.1514
Epoch 2/5
3/3 [=====] - 4s 2s/step - loss: 1.9302 - accuracy: 0.2000 - val_loss: 1.9499 - val_accuracy: 0.1659
Epoch 3/5
3/3 [=====] - 4s 2s/step - loss: 1.8801 - accuracy: 0.2857 - val_loss: 1.9587 - val_accuracy: 0.1659
Epoch 4/5
3/3 [=====] - 4s 2s/step - loss: 1.8177 - accuracy: 0.2857 - val_loss: 1.9511 - val_accuracy: 0.1947
Epoch 5/5
3/3 [=====] - 4s 1s/step - loss: 1.7902 - accuracy: 0.2714 - val_loss: 1.9558 - val_accuracy: 0.1803
```

2. تلخيص النموذج:



3. منحنى الخسارة ومنحنى الدقة (loss & accuracy curves)



إن السبب في الفرق بين المنحنيين هو شكل الصور الإضافية من عملية زيادة البيانات، حيث أنها بقيت بشكل (48, 48, 1)، ولم تأخذ شكل دخل الشبكة الذي هو (224, 224, 3).

النموذج الثاني:

قمنا باعتماد مجموعة البيانات الجزئية التالية:

المفاجأة	الحيادية	الحزن	السعادة	الخوف	الاشمئاز	الغصب	مجموعات التدريب
100	100	100	100	100	100	100	مجموعات التدريب
250	250	250	250	250	250	250	مجموعات الاختبار

النموذج الثاني: قمنا ببناء نموذجنا على شبكة (*EfficientNetB0*), وقمنا باستخراج السمات وقمنا بأخذ أوزان هذه الشبكة وضبط مخرجاتها لتكون ملائمة لمشكلتنا، وأيضاً قمنا بزيادة البيانات عن طريق *(data augmentation)*.

1. تقييم النموذج:

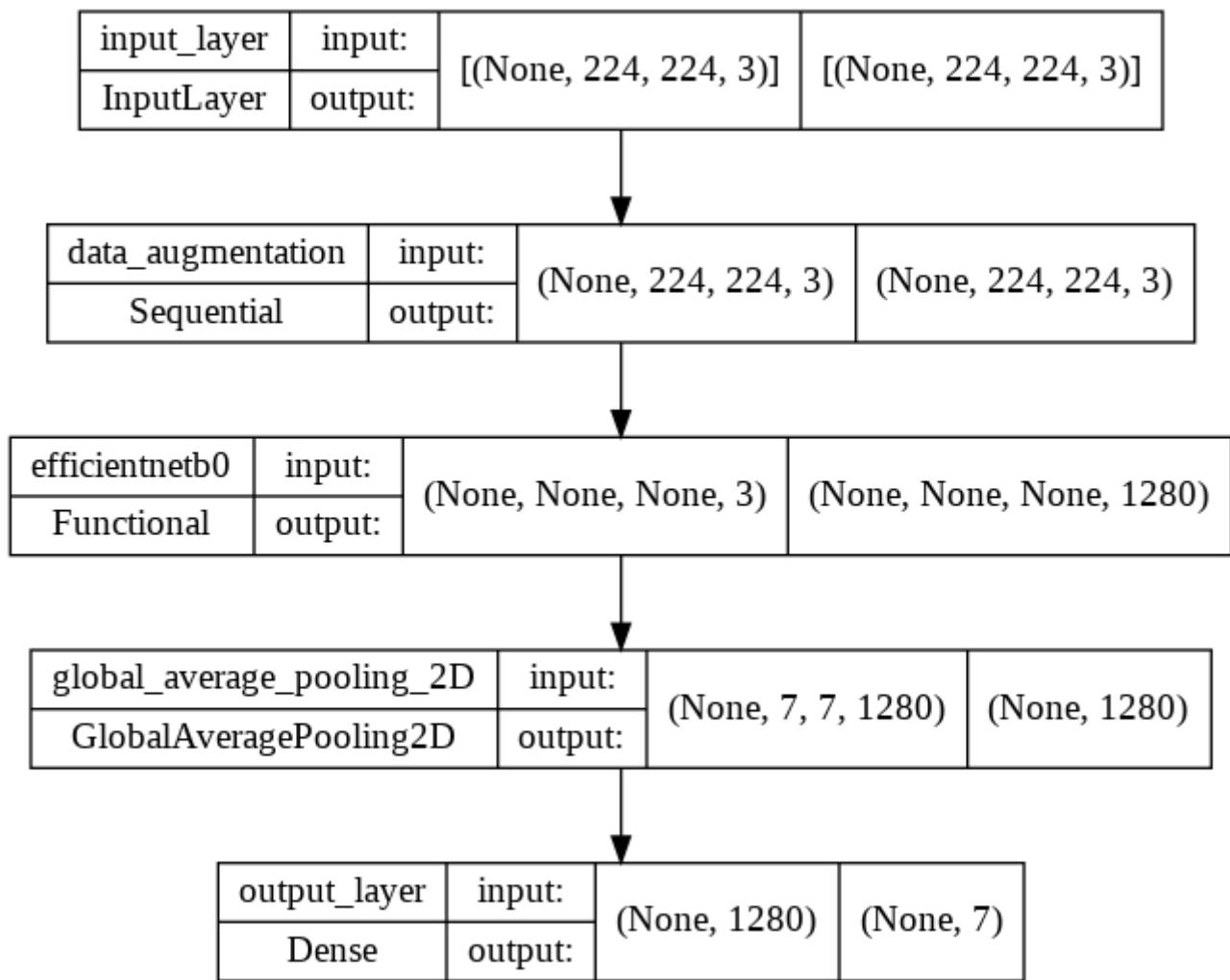
```
[ ] # Check model_2 results on all test_data
results_data_mini_data_aug = model_2.evaluate(test_data)
results_data_mini_data_aug

55/55 [=====] - 8s 131ms/step - loss: 1.7703 - accuracy: 0.3000
[1.7703043222427368, 0.30000001192092896]
```

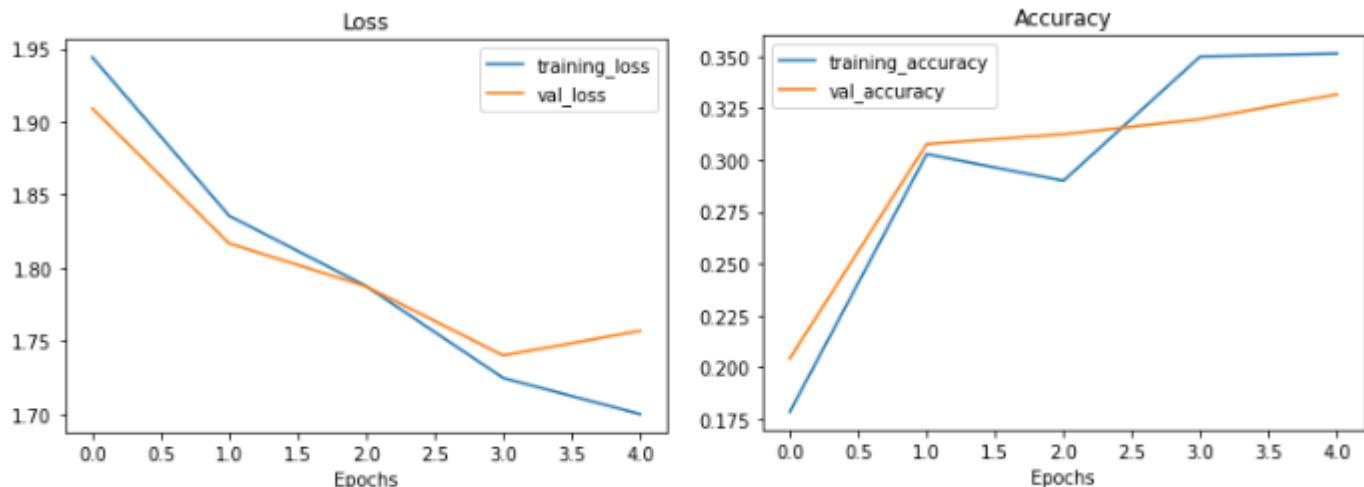
وقد تم تدريب النموذج لـ 5 دورات (*epochs*):

```
Saving TensorBoard log files to: transfer_learning/10_percent_data_aug/20220405-195130
Epoch 1/5
22/22 [=====] - ETA: 0s - loss: 1.9441 - accuracy: 0.1786
Epoch 1: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
22/22 [=====] - 29s 915ms/step - loss: 1.9441 - accuracy: 0.1786 - val_loss: 1.9091 - val_accuracy: 0.2043
Epoch 2/5
22/22 [=====] - ETA: 0s - loss: 1.8357 - accuracy: 0.3029
Epoch 2: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
22/22 [=====] - 12s 535ms/step - loss: 1.8357 - accuracy: 0.3029 - val_loss: 1.8168 - val_accuracy: 0.3077
Epoch 3/5
22/22 [=====] - ETA: 0s - loss: 1.7876 - accuracy: 0.2900
Epoch 3: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
22/22 [=====] - 13s 570ms/step - loss: 1.7876 - accuracy: 0.2900 - val_loss: 1.7874 - val_accuracy: 0.3125
Epoch 4/5
22/22 [=====] - ETA: 0s - loss: 1.7248 - accuracy: 0.3500
Epoch 4: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
22/22 [=====] - 11s 505ms/step - loss: 1.7248 - accuracy: 0.3500 - val_loss: 1.7402 - val_accuracy: 0.3197
Epoch 5/5
22/22 [=====] - ETA: 0s - loss: 1.7001 - accuracy: 0.3514
Epoch 5: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
22/22 [=====] - 12s 546ms/step - loss: 1.7001 - accuracy: 0.3514 - val_loss: 1.7570 - val_accuracy: 0.3317
```

2. تلخيص النموذج:



3. منحني الخسارة ومنحني الدقة (loss & accuracy curves):



النموذج الثالث: هو النموذج الثاني نفسه، لكن تم تدريبه لـ 15 دورة (epochs) بالإضافة لـ 5 دورات التي تم تدريب النموذج الثاني عليها.

1. من أجل عملية الضبط الدقيق، فإننا سوف نقوم بتحديد السبع طبقات الأخيرة من نموذجنا الابتدائي لتصبح قابلة للتدريب.

```
[ ] # To begin fine-tuning, let's start by setting the last 7 layers of our base_model.trainable = True
base_model.trainable = True

# Freeze all layers except for the last 7
for layer in base_model.layers[:-7]:
    layer.trainable = False

# Recompile (we have to recompile our models every time we make a change)
model_2.compile(loss="categorical_crossentropy",
                 optimizer=tf.keras.optimizers.Adam(learning_rate=0.0001), # when fine-tuning you typically want to lower the learning rate by 10x*
                 metrics=["accuracy"])
```

2. نقوم بالتدريب بعد عملية الضبط الدقيق لـ 15 دورة (epochs)

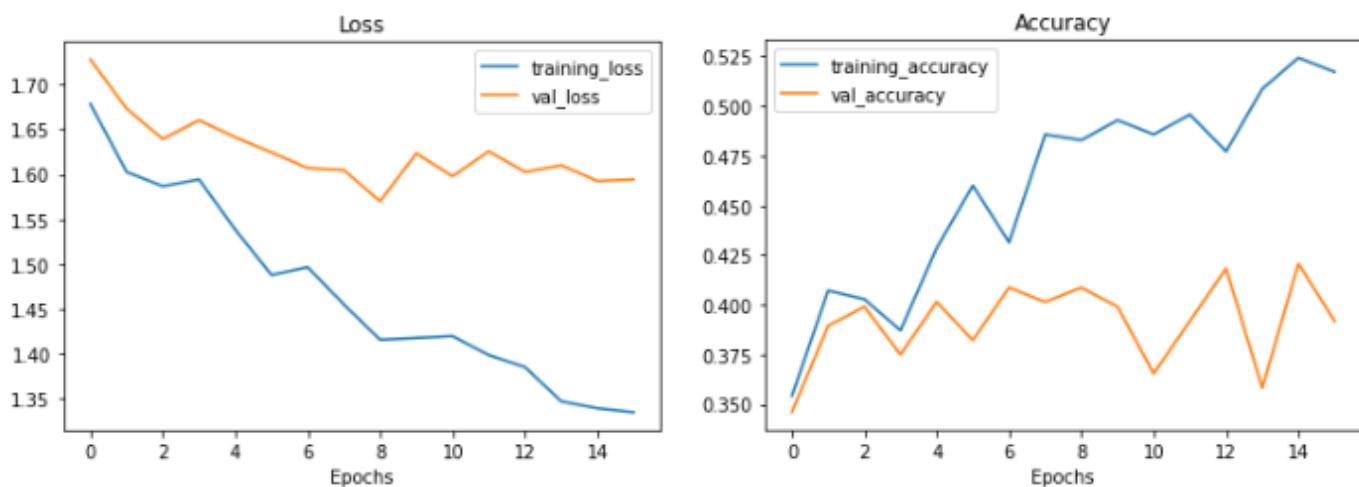
```
Saving TensorBoard log files to: transfer_learning/mini_fine_tune_last_10/20220405-195408
Epoch 5/20
22/22 [=====] - 24s 681ms/step - loss: 1.6782 - accuracy: 0.3543 - val_loss: 1.7277 - val_accuracy: 0.3462
Epoch 6/20
22/22 [=====] - 12s 530ms/step - loss: 1.6026 - accuracy: 0.4071 - val_loss: 1.6730 - val_accuracy: 0.3894
Epoch 7/20
22/22 [=====] - 12s 550ms/step - loss: 1.5867 - accuracy: 0.4029 - val_loss: 1.6392 - val_accuracy: 0.3990
Epoch 8/20
22/22 [=====] - 10s 474ms/step - loss: 1.5943 - accuracy: 0.3871 - val_loss: 1.6601 - val_accuracy: 0.3750
Epoch 9/20
22/22 [=====] - 12s 542ms/step - loss: 1.5384 - accuracy: 0.4286 - val_loss: 1.6411 - val_accuracy: 0.4014
Epoch 10/20
22/22 [=====] - 12s 524ms/step - loss: 1.4880 - accuracy: 0.4600 - val_loss: 1.6243 - val_accuracy: 0.3822
Epoch 11/20
22/22 [=====] - 11s 500ms/step - loss: 1.4967 - accuracy: 0.4314 - val_loss: 1.6069 - val_accuracy: 0.4087
Epoch 12/20
22/22 [=====] - 11s 496ms/step - loss: 1.4551 - accuracy: 0.4857 - val_loss: 1.6046 - val_accuracy: 0.4014
Epoch 13/20
22/22 [=====] - 11s 473ms/step - loss: 1.4159 - accuracy: 0.4829 - val_loss: 1.5699 - val_accuracy: 0.4087
Epoch 14/20
22/22 [=====] - 11s 507ms/step - loss: 1.4179 - accuracy: 0.4929 - val_loss: 1.6232 - val_accuracy: 0.3990
Epoch 15/20
22/22 [=====] - 9s 421ms/step - loss: 1.4199 - accuracy: 0.4857 - val_loss: 1.5977 - val_accuracy: 0.3654
Epoch 16/20
22/22 [=====] - 11s 493ms/step - loss: 1.3988 - accuracy: 0.4957 - val_loss: 1.6255 - val_accuracy: 0.3918
Epoch 17/20
22/22 [=====] - 9s 425ms/step - loss: 1.3856 - accuracy: 0.4771 - val_loss: 1.6025 - val_accuracy: 0.4183
Epoch 18/20
22/22 [=====] - 10s 459ms/step - loss: 1.3474 - accuracy: 0.5086 - val_loss: 1.6100 - val_accuracy: 0.3582
Epoch 19/20
22/22 [=====] - 11s 482ms/step - loss: 1.3396 - accuracy: 0.5243 - val_loss: 1.5925 - val_accuracy: 0.4207
Epoch 20/20
22/22 [=====] - 9s 396ms/step - loss: 1.3353 - accuracy: 0.5171 - val_loss: 1.5943 - val_accuracy: 0.3918
```

3. تقييم النموذج:

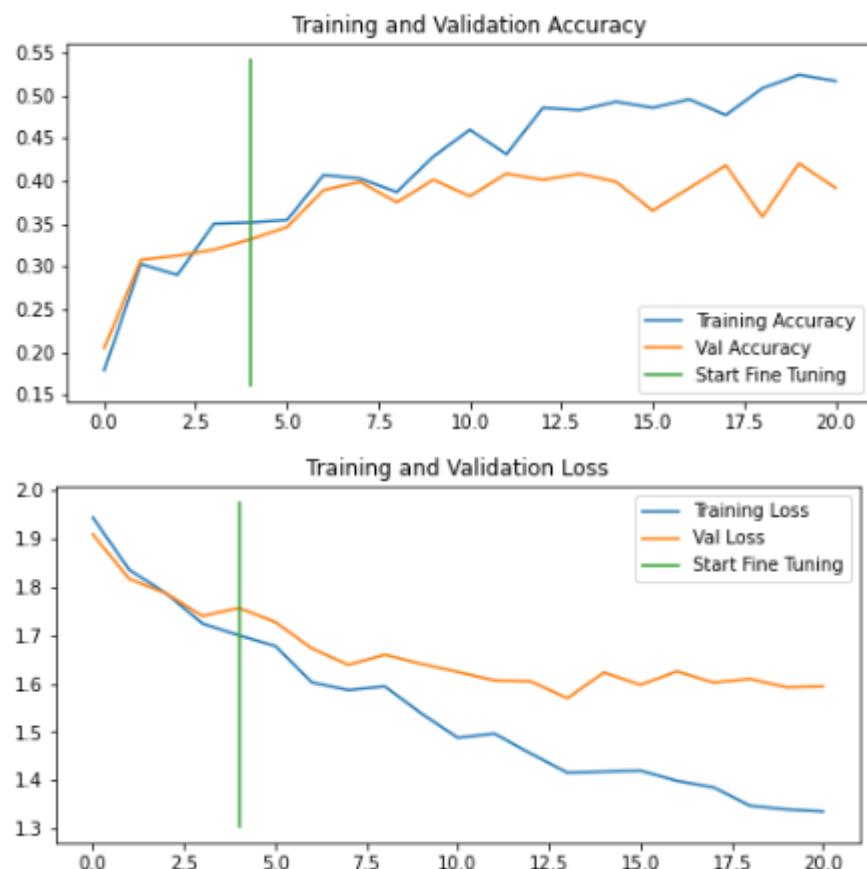
```
[ ] # Evaluate the fine-tuned model (model_3 which is actually model_2 fine-tuned for another 15 epochs)
results_fine_tune_mini = model_2.evaluate(test_data)

55/55 [=====] - 7s 131ms/step - loss: 1.6071 - accuracy: 0.3783
```

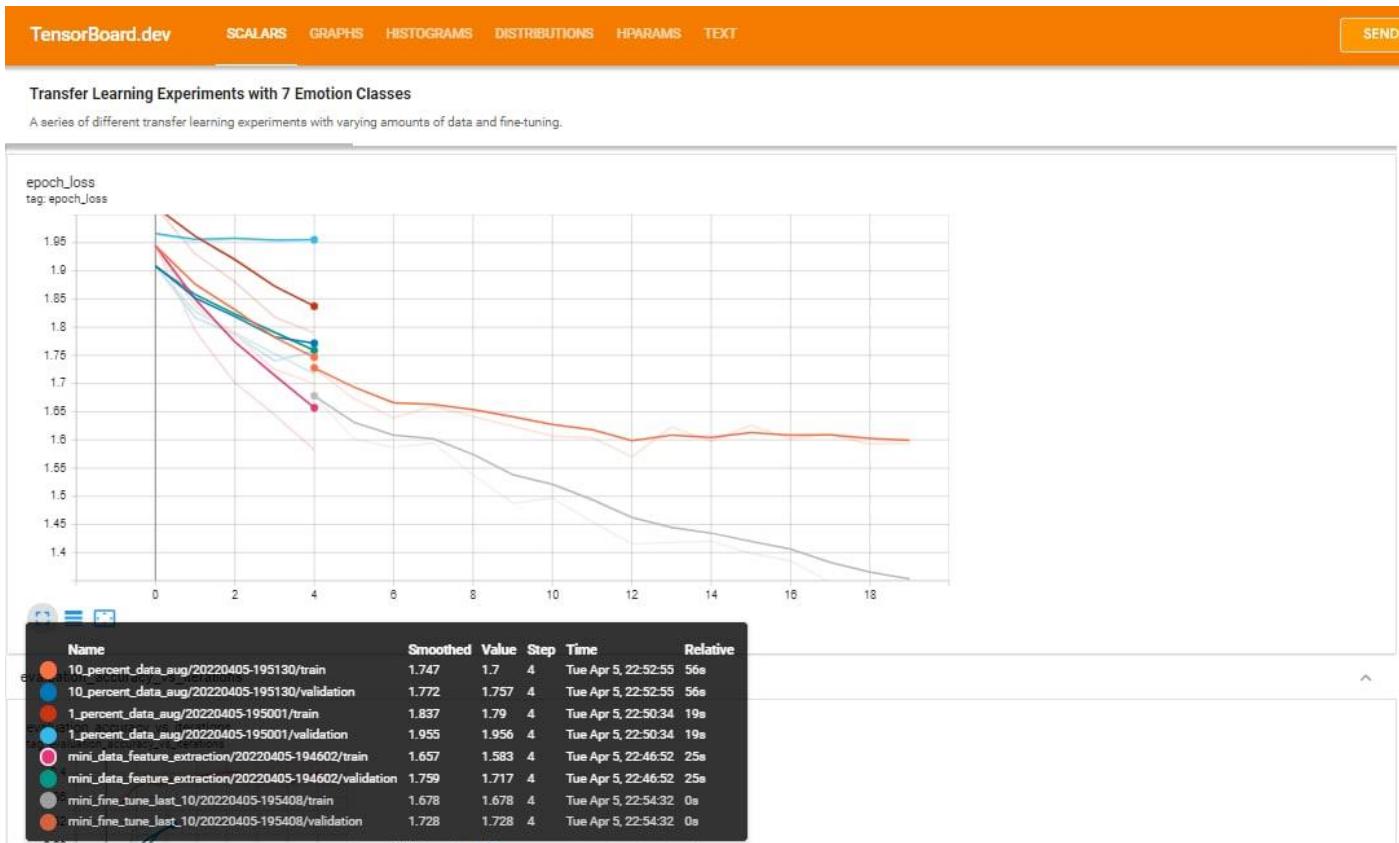
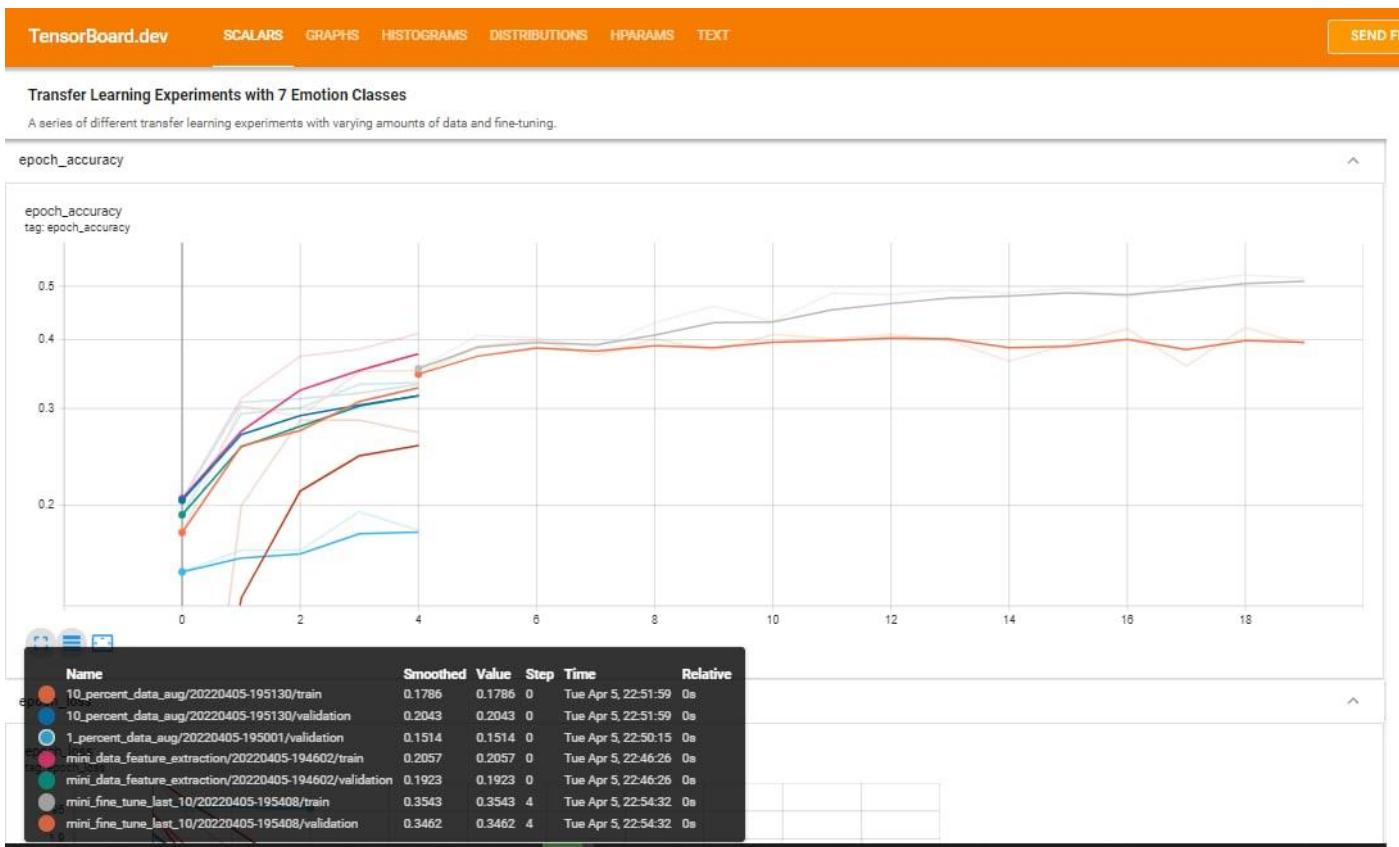
4. منحنى الخسارة ومنحنى الدقة :(*loss & accuracy curves*)

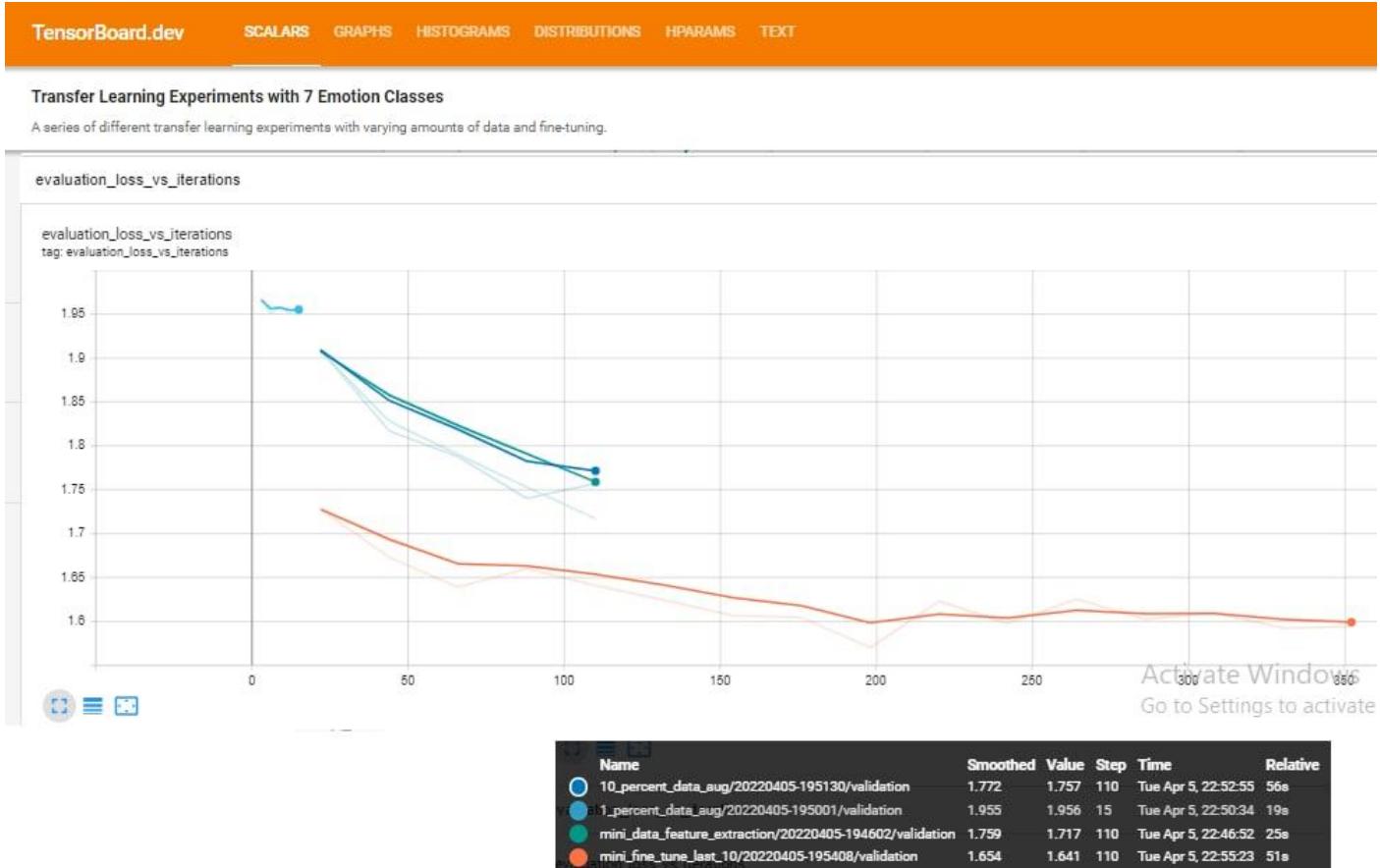
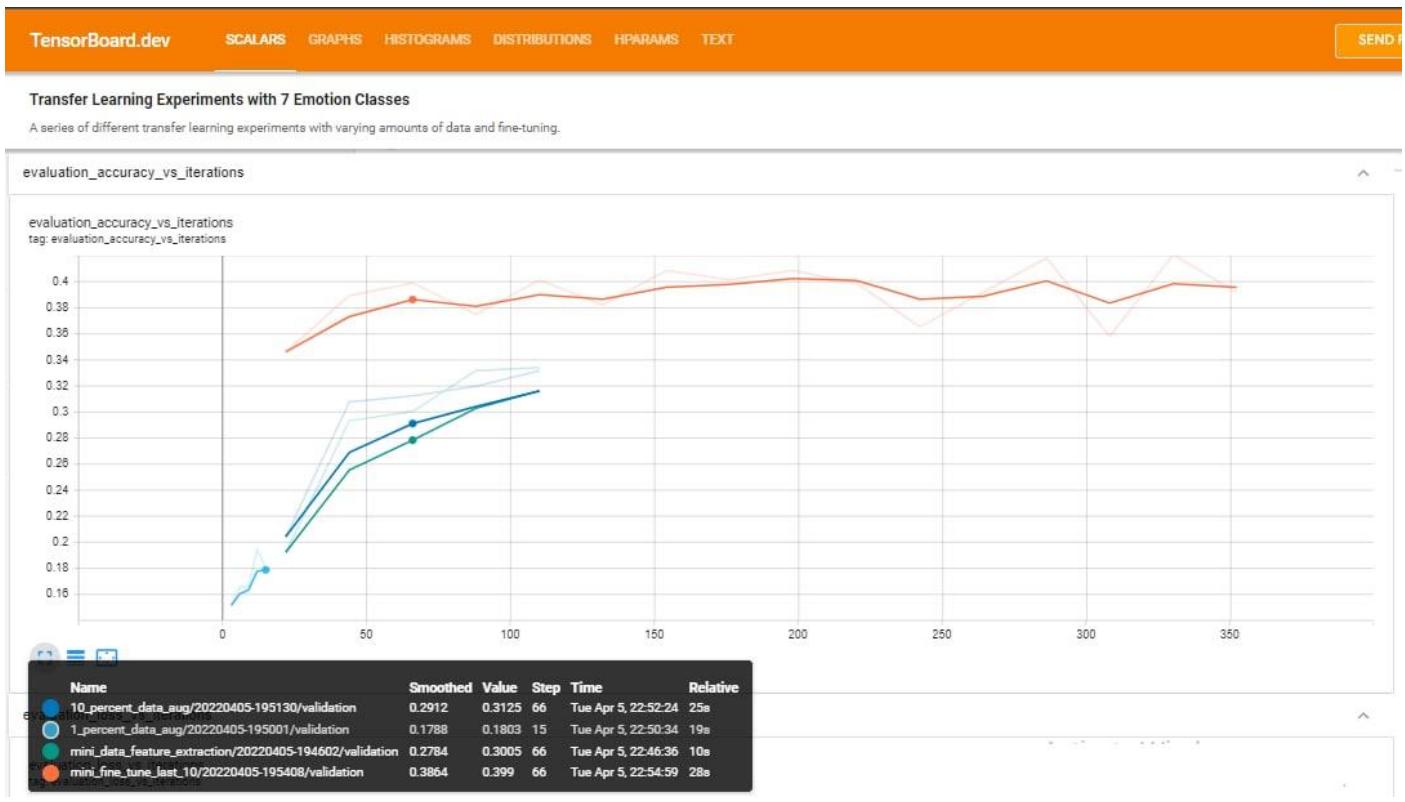


5. من أجل مقارنة نموذجنا مع وبدون الضبط الدقيق، كالتالي:



أما من أجل مقارنة جميع نماذجنا في هذا الملف، فقد تم استعمال (*Tensorboard*)





من أجل الوضوح أكثر، يرجى زيارة الرابط التالي (يستحسن تشغيل VPN):

<https://tensorboard.dev/experiment/trhBeoibQn6lG95nx24QdA/>

انتهى الملف الرابع

الملف الخامس

قمنا باعتماد مجموعة بيانات الصور كاملة، كالتالي:

المفاجأة	الحزن	السعادة	الخوف	الاشمئizar	الغضب	مجموعات التدريب
الحيادية						مجموعات الاختبار
3171	4965	4830	7215	4097	436	3995
831	1233	1247	1774	1024	111	958

إن هذا النموذج هو النموذج الأخير الذي قمنا باعتماده، وتجربته بالزمن الحقيقي.

1. تقييم النموذج:

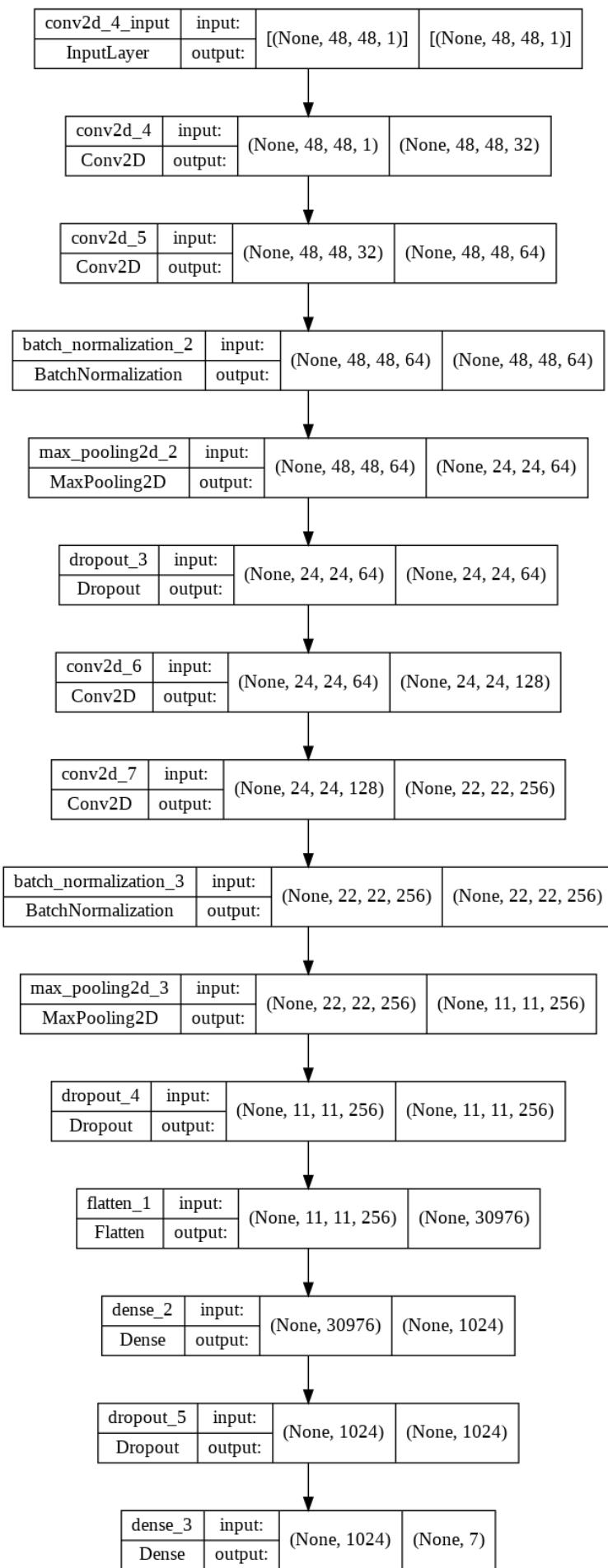
```
[ ] # Evaluate the loaded model
FinalModel.evaluate(training_set), FinalModel.evaluate(test_set)

449/449 [=====] - 27s 59ms/step - loss: 0.5321 - accuracy: 0.8529
113/113 [=====] - 4s 37ms/step - loss: 1.0768 - accuracy: 0.6578
([0.5321141481399536, 0.8529032468795776],
 [1.0768322944641113, 0.6578434109687805])
```

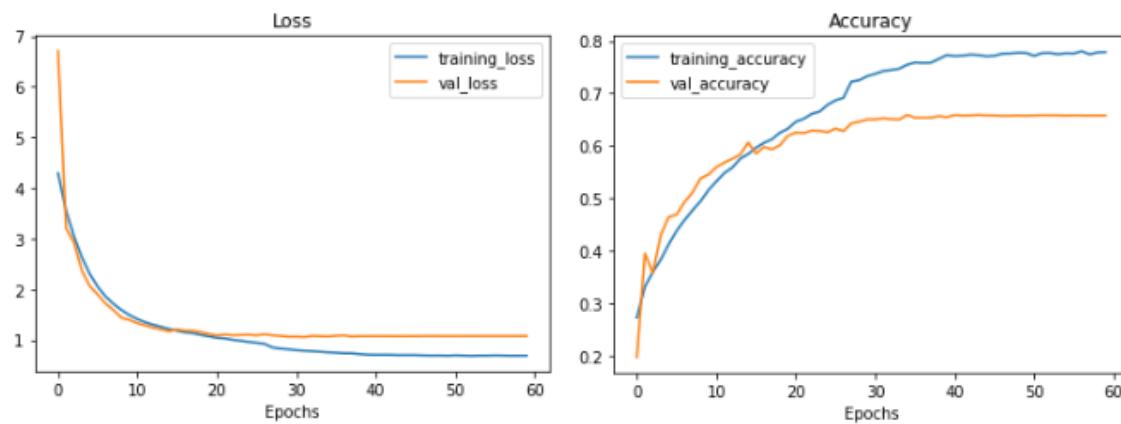
وتم تدريب هذا النموذج لـ 60 دورة، (آخر 5 دورات):

```
Epoch 56: ReduceLROnPlateau reducing learning rate to 3.199999980552093e-08.
448/448 [=====] - 36s 81ms/step - loss: 0.6955 - accuracy: 0.7761 - val_loss: 1.0774 - val_accuracy: 0.6581 - lr: 1.6000e-07
Epoch 57/60
448/448 [=====] - ETA: 0s - loss: 0.6933 - accuracy: 0.7803
Epoch 57: val_loss did not improve from 1.05849
448/448 [=====] - 36s 81ms/step - loss: 0.6933 - accuracy: 0.7803 - val_loss: 1.0770 - val_accuracy: 0.6579 - lr: 3.2000e-08
Epoch 58/60
448/448 [=====] - ETA: 0s - loss: 0.6894 - accuracy: 0.7744
Epoch 58: val_loss did not improve from 1.05849
448/448 [=====] - 37s 82ms/step - loss: 0.6894 - accuracy: 0.7744 - val_loss: 1.0773 - val_accuracy: 0.6576 - lr: 3.2000e-08
Epoch 59/60
448/448 [=====] - ETA: 0s - loss: 0.6897 - accuracy: 0.7781
Epoch 59: val_loss did not improve from 1.05849
448/448 [=====] - 37s 82ms/step - loss: 0.6897 - accuracy: 0.7781 - val_loss: 1.0763 - val_accuracy: 0.6579 - lr: 3.2000e-08
Epoch 60/60
448/448 [=====] - ETA: 0s - loss: 0.6909 - accuracy: 0.7787
Epoch 60: val_loss did not improve from 1.05849
448/448 [=====] - 36s 80ms/step - loss: 0.6909 - accuracy: 0.7787 - val_loss: 1.0773 - val_accuracy: 0.6578 - lr: 3.2000e-08
```

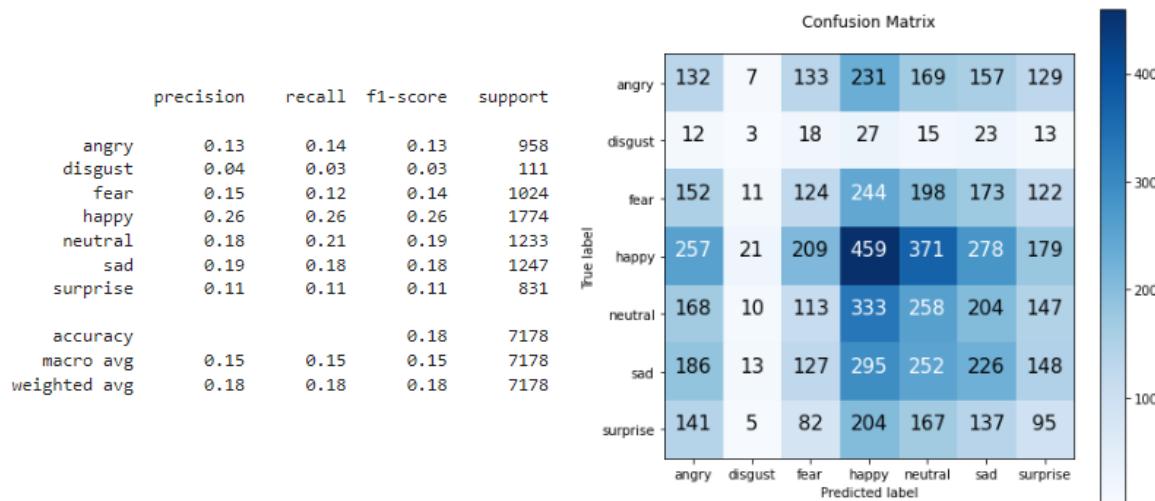
2. تلخيص النموذج:



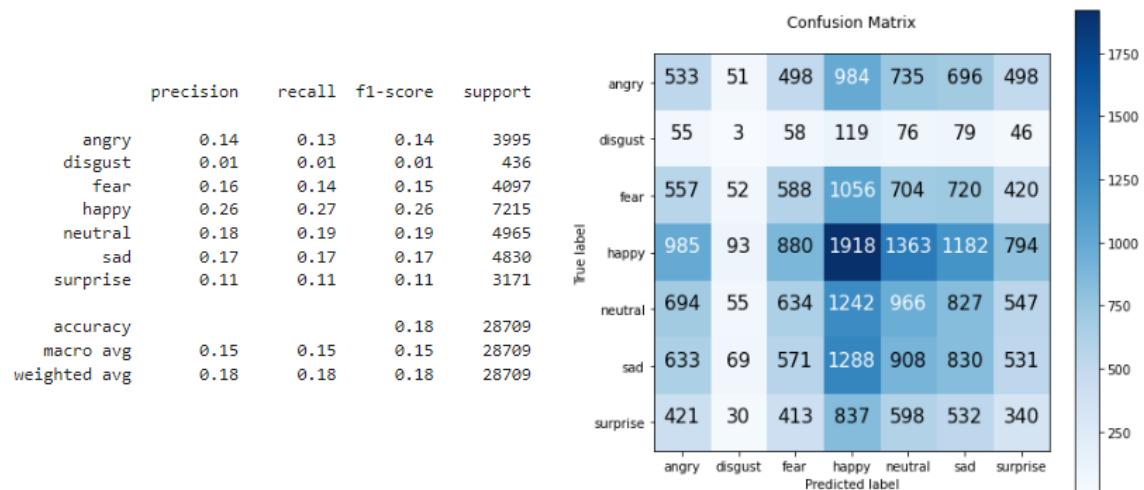
3. منحنى الخسارة ومنحنى الدقة (loss & accuracy curves)



4. مصفوفة الاختلاط (confusion matrix) و报 告 (classification report) لمجموعات الاختبار:



5. مصفوفة الاختلاط (confusion matrix) و报 告 (classification report) لمجموعات التدريب:



الفصل السادس

شرح الاستخدام

من أجل تجربة النموذج، نحتاج لجهاز حاسب مثبت عليه (*Anaconda Software*) بالإضافة إلى تثبيت مكتبة (*OpenCV*) وتنشيط بيئة (*Jupyter Notebook*) داخل بيئة (.tf).

مرفق داخل مجلد المشروع، مجلد باسم (*test*), يحوي بداخله النموذج النهائي بحجم (367 MB)، بالإضافة إلى ملف (*haarcascade_frontalface_default*)، الذي يساعد في تحديد مكان الوجه في الصورة من خلال مكتبة (*OpenCV*).

```
In [1]: 1 import numpy as np
2 import tensorflow as tf
3 from tensorflow.keras.models import load_model
4 import cv2
5 from tensorflow.keras.preprocessing import image
6 from tensorflow.keras.preprocessing.image import img_to_array
7 from time import sleep
```

نقوم بتضمين المكتبات التي تحتاجها من أجل التعامل مع النموذج بشكل عام.

```
In [ ]: 1 face_classifier = cv2.CascadeClassifier(r'haarcascade_frontalface_default.xml')
2 classifier = load_model(r'FinalModel.h5')
3
4 emotion_labels = ['Angry', 'Disgust', 'Fear', 'Happy', 'Neutral', 'Sad', 'Surprise']
5
6 cap = cv2.VideoCapture(0)
7
8
9 while True:
10     _, frame = cap.read()
11     labels = []
12     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
13
14     faces = face_classifier.detectMultiScale(gray)
15
16     for (x,y,w,h) in faces:
17         cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,255),2)
18         roi_gray = gray[y:y+h,x:x+w]
19         roi_gray = cv2.resize(roi_gray,(48,48),interpolation=cv2.INTER_AREA)
20
21
22         if np.sum([roi_gray])!=0:
23             roi = roi_gray.astype('float')/255.0
24             roi = img_to_array(roi)
25             roi = np.expand_dims(roi,axis=0)
26
27             prediction = classifier.predict(roi)[0]
28             label=emotion_labels[prediction.argmax()]
29             label_position = (x,y-10)
30             cv2.putText(frame,label,label_position,cv2.FONT_HERSHEY_SIMPLEX,1,(0,255,0),2)
31
32         else:
33             cv2.putText(frame,'No Faces',(30,80),cv2.FONT_HERSHEY_SIMPLEX,1,(0,255,0),2)
34
35         cv2.imshow('Emotion Detector',frame)
36         if cv2.waitKey(1) & 0xFF == ord('q'):
37             break
38
39 cap.release()
40 cv2.destroyAllWindows()
```

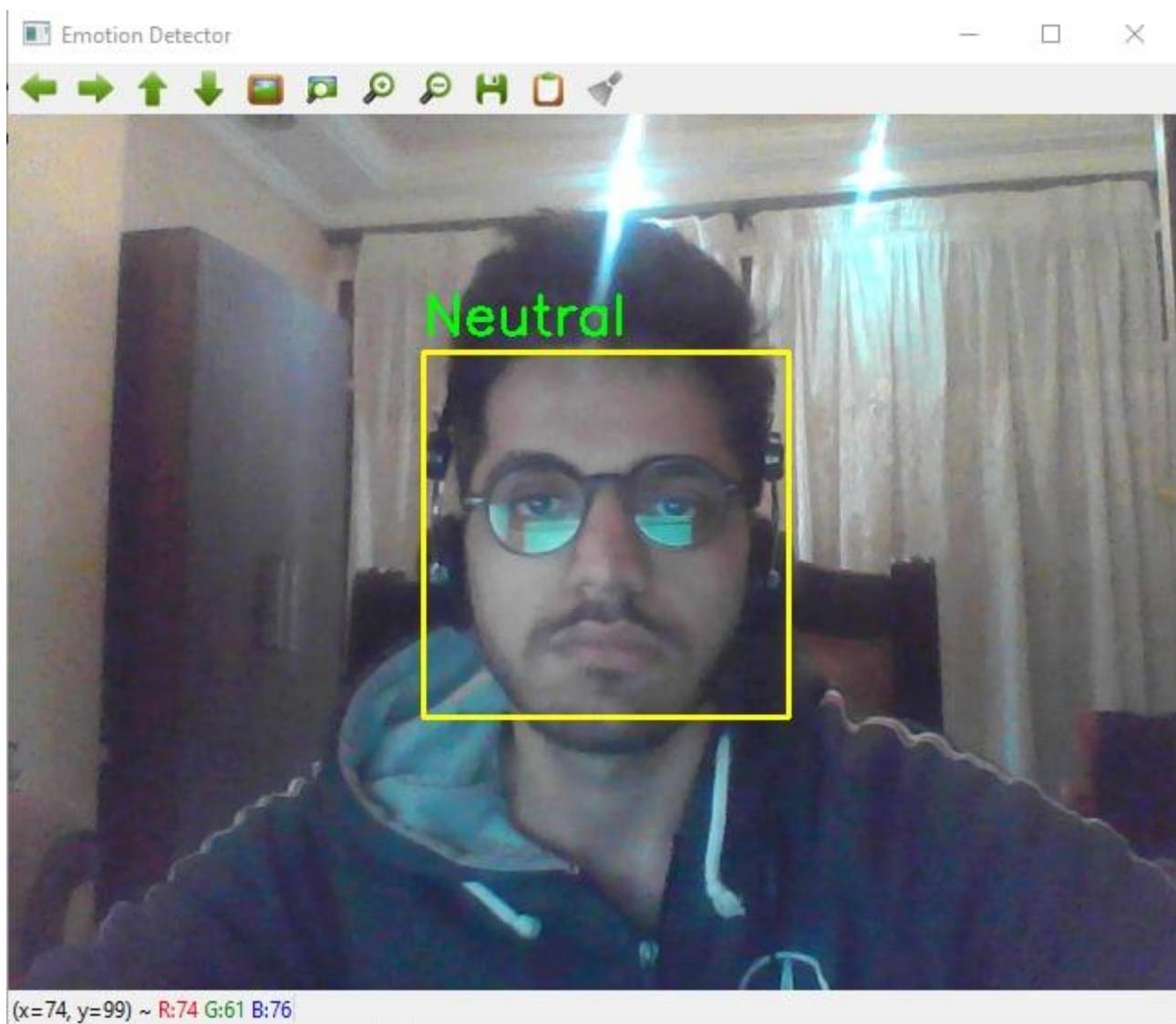
شرح البرنامج:

نقوم بتضمين ملف (haarcascade_frontalface_default) الذي يساعد بتحديد مكان الوجه، ونقوم بتضمين النموذج، ثم نقوم بإعادة فيديو مباشر من كاميرا الحاسب المستخدم.

ثم نقوم بتحويل الصورة المأخوذة إلى اللون الرمادي، ونقوم بتحديد مكان الوجه في الصورة، وبعدها نقوم برسم مستطيل حول الوجه في على الكاميرا، ومن ثم نقوم باقتطاع صورة الوجه وحفظها في متغير جديد (*roi_gray*) مستطيل حول الوجه في على الكاميرا، ومن ثم نقوم باقتطاع صورة الوجه وحفظها في متغير جديد (*roi*: Region Of Interest)، ثم بعدها نقوم بتغيير أبعاد الصورة لكي تتناسب مع دخل النموذج الخاص بنا (Normalization). وبعدها نقوم بـ (Normalization) لصورة الوجه المأخوذة، ونقوم بتحويلها لمصفوفة من نوع (Numpy). وبعدها نقوم بالتنبؤ بالشعور الموجود في الصورة ونقوم بطباعته أعلى مستطيل تحديد الوجه على الكاميرا مباشرة.

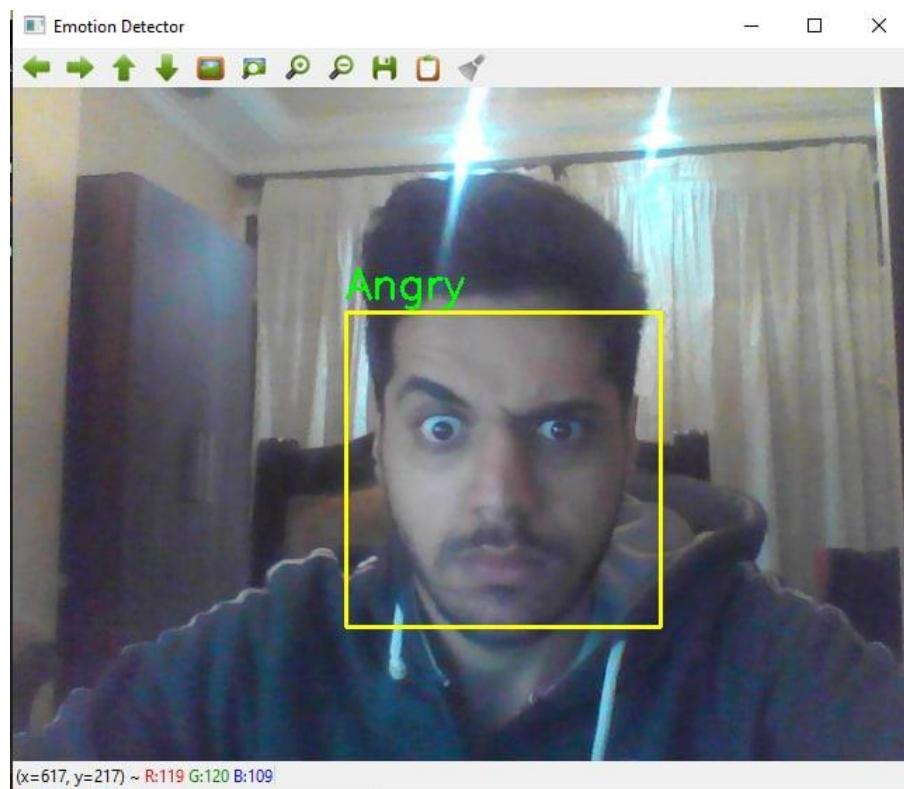
للخروج من البرنامج، يمكن ضغط زر (*q*) مرتان.

عند تشغيل البرنامج، يقوم بفتح الكاميرا وأول خرج على الكاميرا، يكون كالتالي:

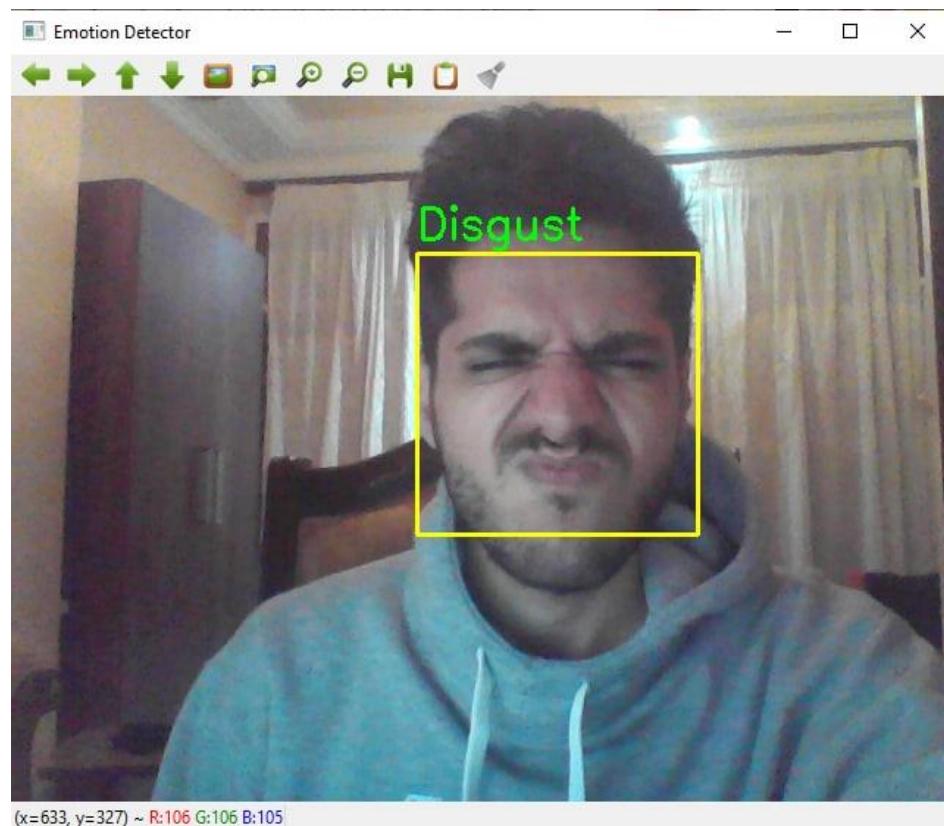


تجربة المشاعر السابعة:

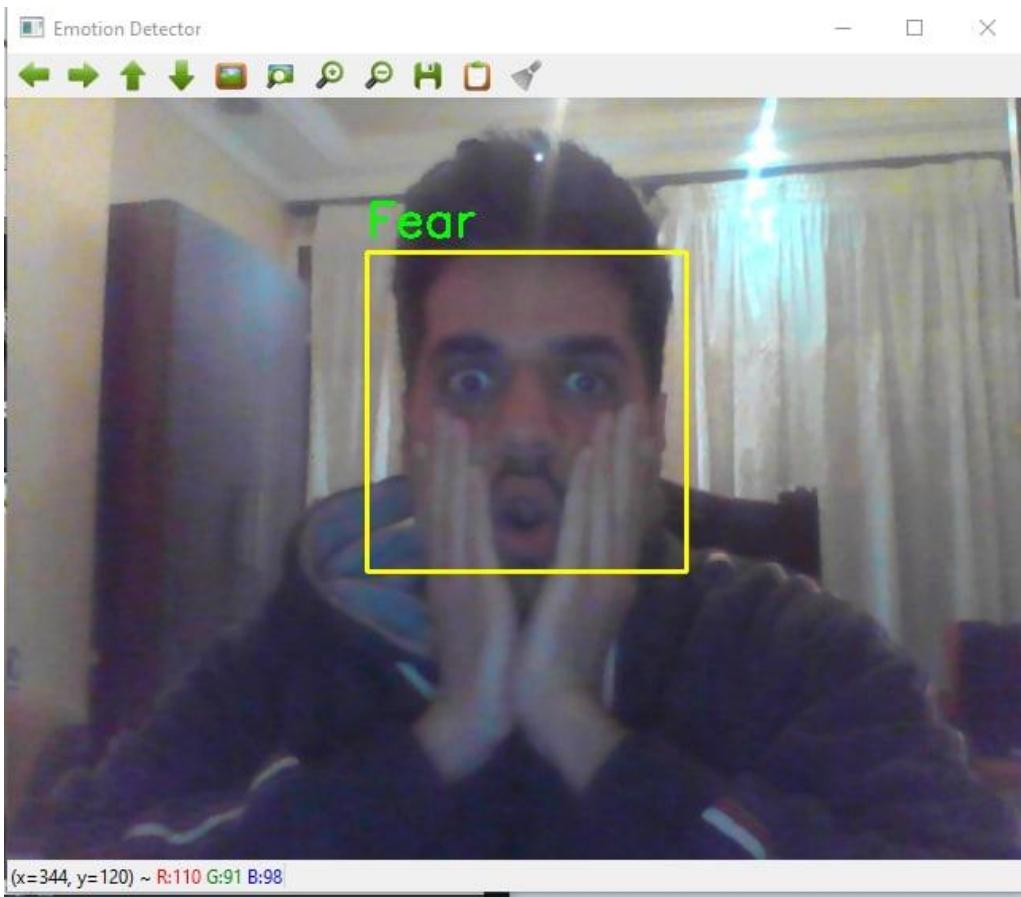
1. الغضب:



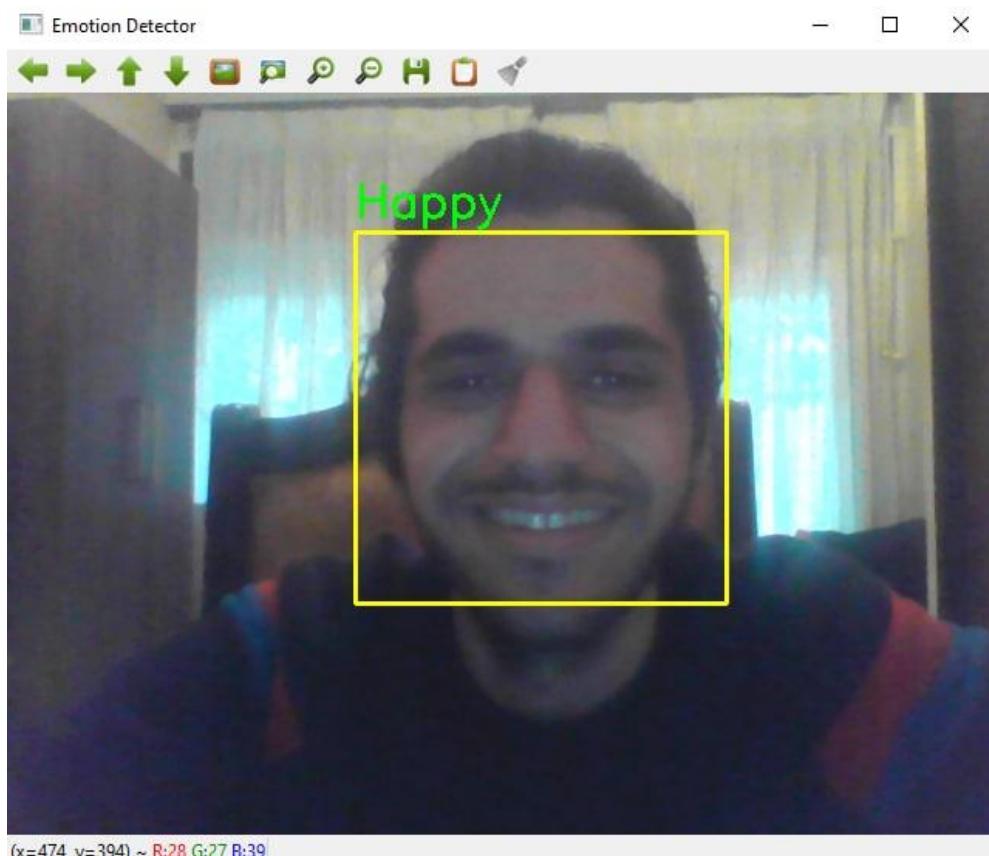
2. الاشمئاز:



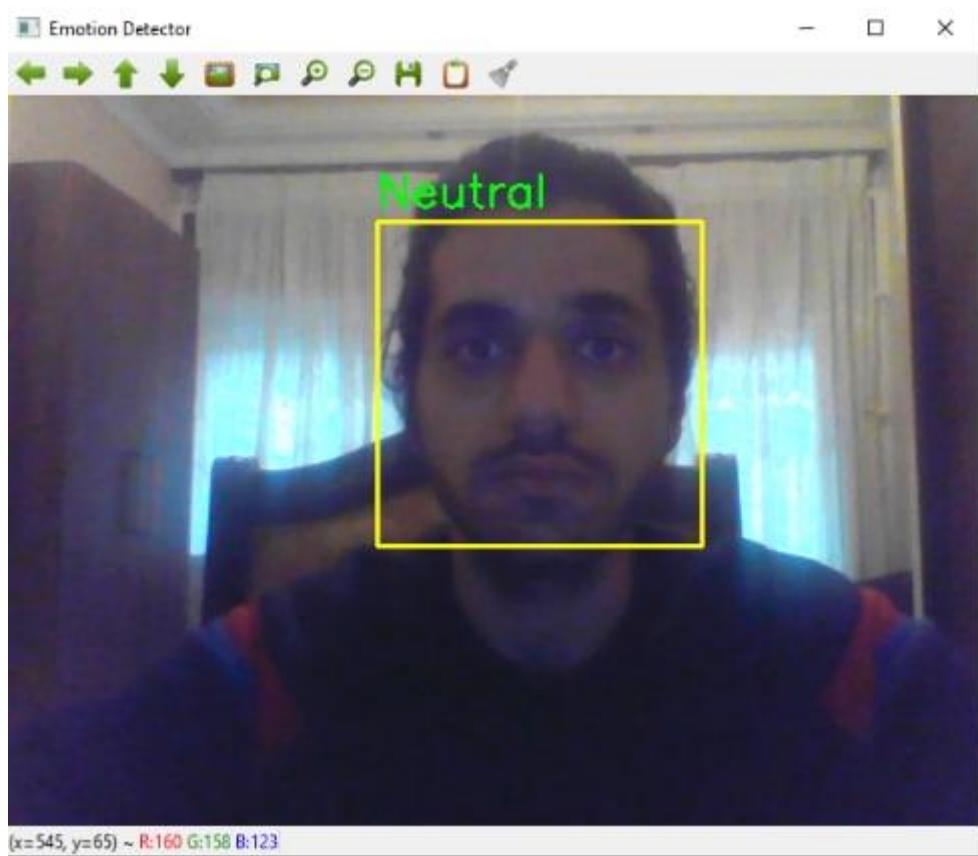
الخوف: 3



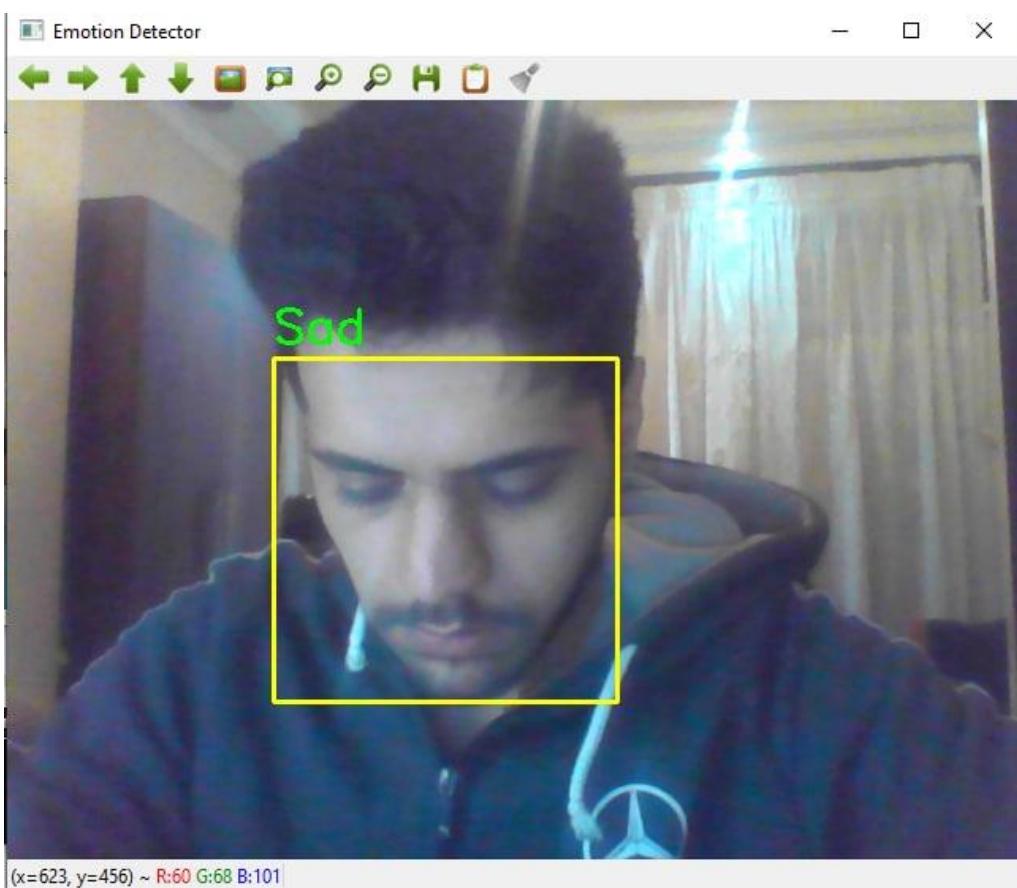
السعادة: 4

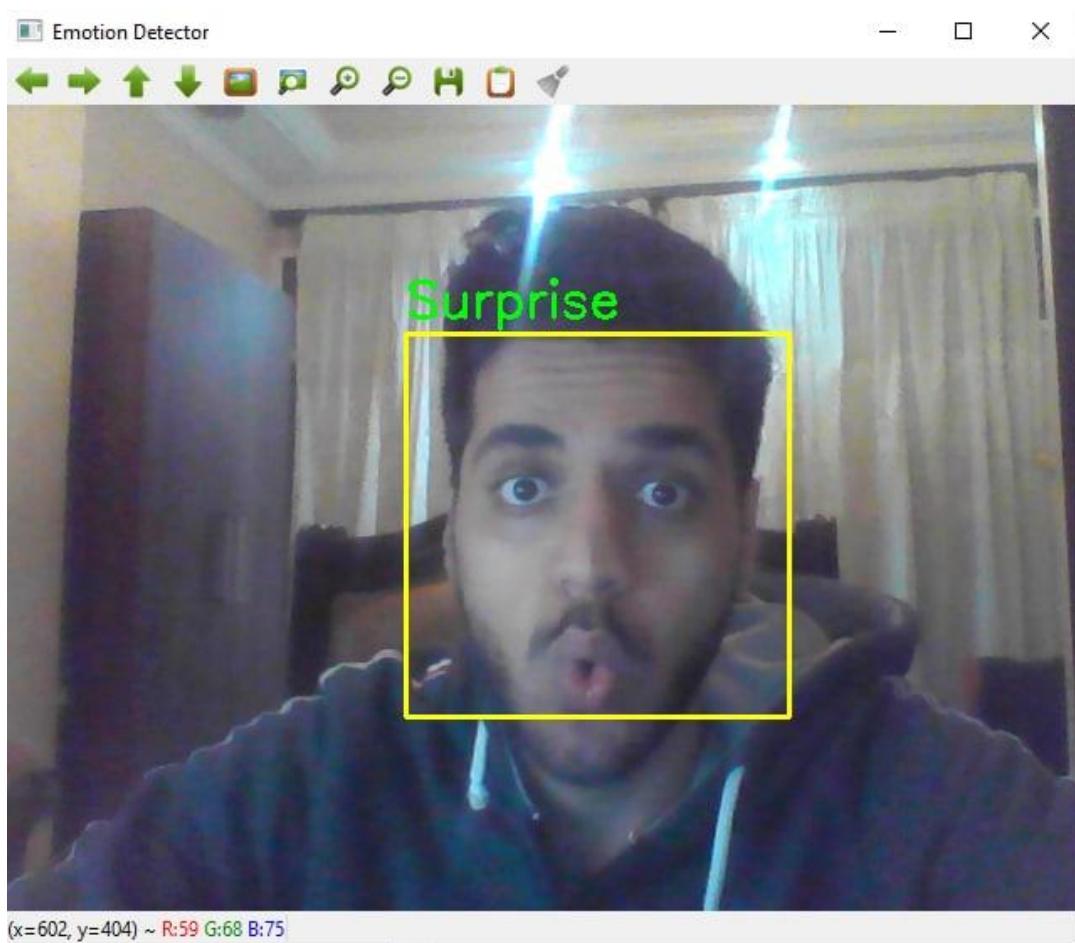


5. الحيادية:

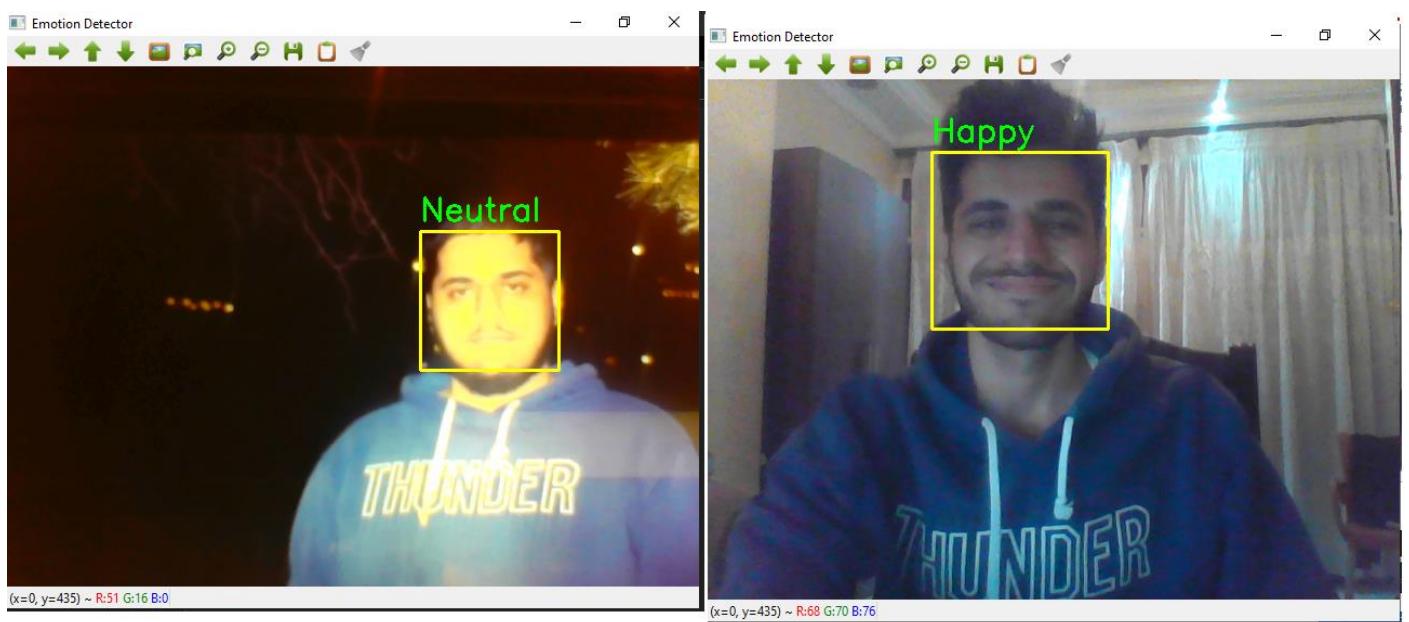


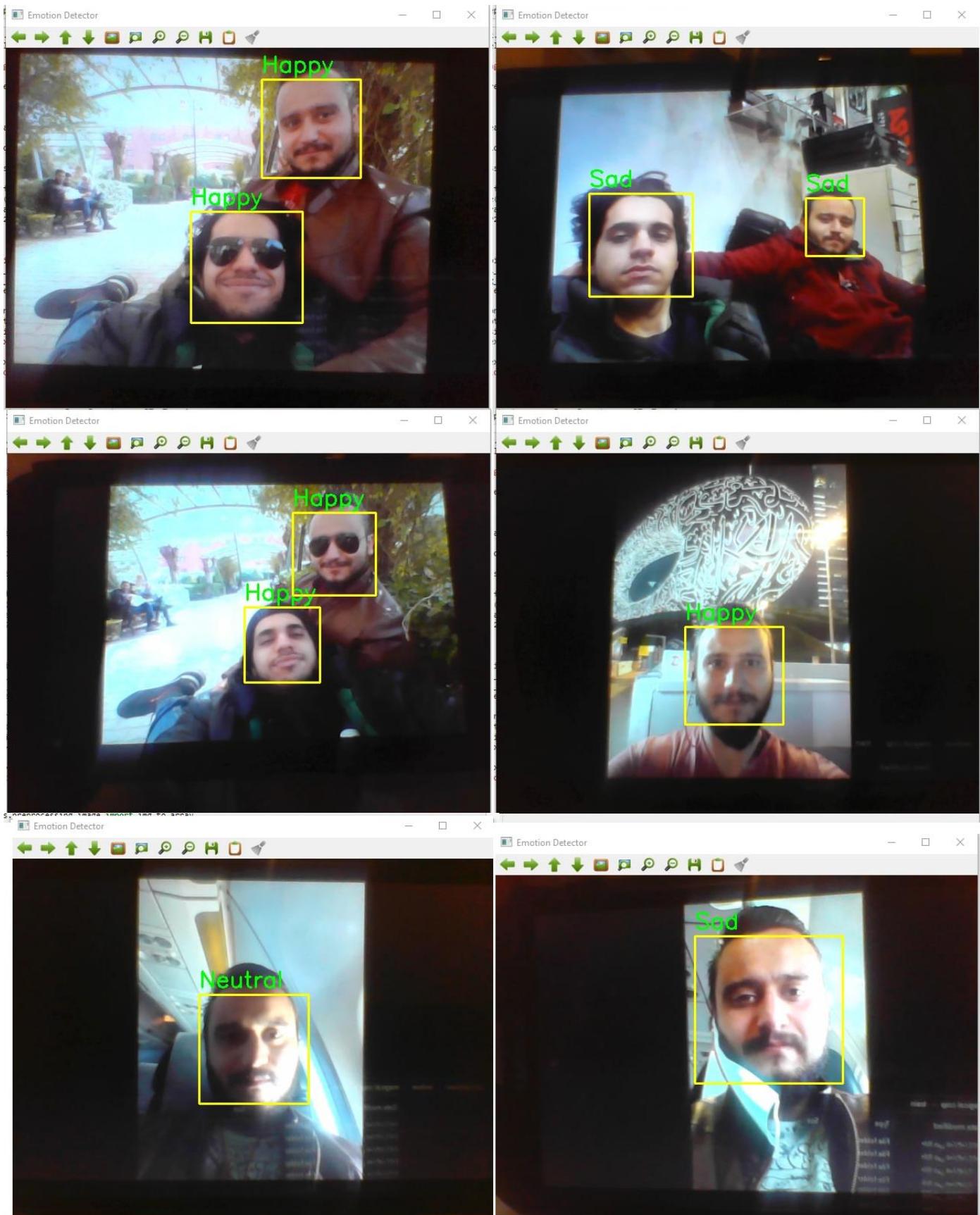
6. الحزن:





خرج إضافي للتجربة فقط:





الفصل السابع

الخاتمة

الخاتمة:

في النهاية، في عصرنا الحالي تتقدم نظم الذكاء الاصطناعي لأنظمة التعرف على المشاعر وغيرها بشكل كبير، بسبب ثورة التكنولوجيا التي يشهدها هذا القرن، وإن ما تم ذكره وتنفيذه من نماذج في هذا المشروع ما هو إلا جزء صغير من علم كامل في التعرف على مشاعر الإنسان في مجال الذكاء الاصطناعي والتعلم العميق وتعلم الآلة.

مقارنة لكل النماذج التي قمنا بتجربتها:

الملف الأول: 1. CNNsBinaryClassification						
النموذج	مجموعات الصور	مجموعات بيانات	الدقة على التدريب	الدقة على المجموعات	الدقة على المجموعات	الضياع على الاختبار
الأول	الحزن والسعادة	10 epochs	0.8584	0.3087	0.8375	0.3652
الثاني	الحزن والسعادة	10 epochs	0.7230	0.5562	0.7219	0.5584
الثالث	الحزن والسعادة	10 epochs	0.7279	0.5592	0.7297	0.5599
الرابع	الحزن والسعادة	10 epochs	0.9359	0.1872	0.7809	0.5033
الخامس	الحزن والسعادة	10 epochs	0.8363	0.3656	0.8160	0.4006
السادس	الحزن والسعادة	10 epochs	0.6568	0.6391	0.6640	0.6356
السابع	الحزن والسعادة	10 epochs	0.7460	0.5369	0.7276	0.5512
الملف الثاني: 2. CNNsMultiClassification						
الأول	كاملة	5 epochs	0.5238	1.2682	0.4819	1.3468
الثاني	كاملة	5 epochs	0.5049	1.3290	0.4724	1.3934
الثالث	كاملة	5 epochs	0.3826	1.5684	0.3799	1.5672
الملف الثالث:						
3. Feature Extraction Experiments With Resnet50 & EfficientNetB0						
الأول	صورة 100 للتدريب لكل نموذج	10 epochs	0.7829	0.8280	0.3749	1.7018
	صورة 250 اختبار لكل نموذج					
الثاني	صورة 100 للتدريب لكل نموذج	10 epochs	0.5414	1.3583	0.3440	1.6937
	صورة 250 اختبار لكل نموذج					

الملف الرابع: TransferLearningFeatureExtraction&FineTuning						
1. 7218	0. 3297	—	—	5 epochs	100 صورة للتدريب لكل نموذج	الابتدائي
					250 صورة اختبار لكل نموذج	
1. 9592	0. 1834	—	—	5 epochs	10 صور للتدريب لكل نموذج	الأول
					250 صورة اختبار لكل نموذج	
1. 7703	0. 30	—	—	5 epochs	100 صورة للتدريب لكل نموذج	الثاني
					250 صورة اختبار لكل نموذج	
1. 6071	0. 3783	—	—	15 epochs	100 صورة للتدريب لكل نموذج	الثالث
					250 صورة اختبار لكل نموذج	
الملف الخامس: FINALCNNFER						
1. 0768	0. 6578	0. 5321	0. 8529	60 epochs	كاملة	النموذج الأخير

الآفاق المستقبلية:

إن عملية بناء نماذج التعلم العميق هي ليس بالإمر السهل فهو يتطلب أن يكون ملماً بعلم الإحصاء والاحتمالات والبرمجة والتفكير المنطقي وإدارة المشاريع والتحليل والتصميم للوصول للفكرة أو التطبيق المنشود.

1-2 إمكانيات التحسين والتطوير المستقبلية على مستوى الدراسة:

يعتبر هذا التطوير تطويراً على المستوى القاعدي (المستوى الأول) الذي هو النموذج، ويمكن أن يتم بالمحاور الأساسية التالية:

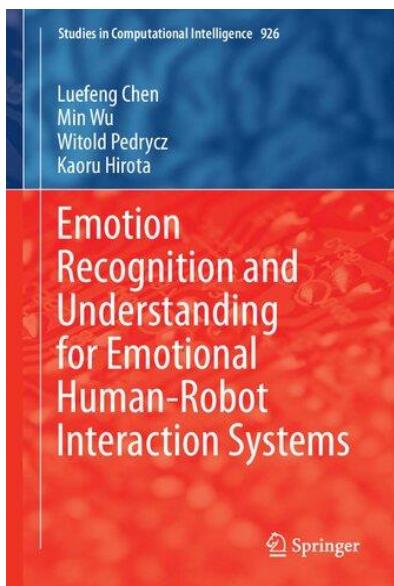
تطوير هذا النموذج بشكل صحيح ليحقق دقة أعلى.

جعل النموذج ديناميكياً قابلاً للتوسيع ليشمل أكثر عدد ممكن من المشاعر.

تطوير تطبيق للأجهزة الخليوية من أجل التعرف على المشاعر

2-2 إمكانيات التحسين والتطوير المستقبلية على المستوى الأعلى:

توسيعة قاعدة النظام بأكمله لجعله يشمل التعرف على عمر الشخص و الجنس، وحالته الذهنية مثلاً.

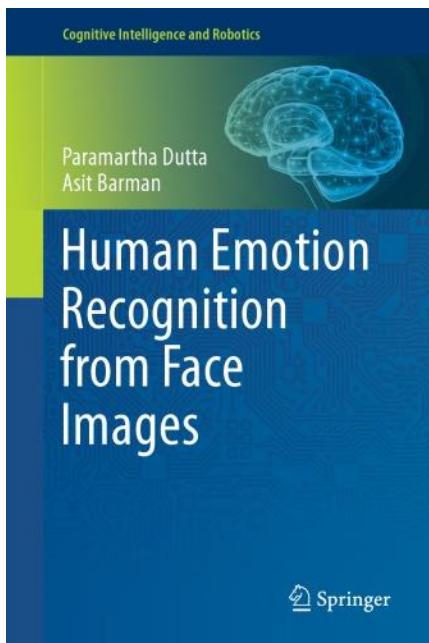


1. Emotion Recognition and Understanding for Emotional Human-Robot Interaction Systems.

Authors: Kaoru Hirota, Min Wu, Athanasios V. Vasilakos, Luefeng Chen

Originally published: November 13, 2020.

Link: <http://library.lol/main/76E33713199AF59511C871E202AF7678>



2. Human Emotion Recognition from Face Images

Authors: Asit Barman, Paramartha Dutta

Originally published: March 26, 2020.

Link: <http://library.lol/main/1D19A0383EFE7822A2CB8532F2DE9124>

3. FER-2013 DATASET ON KAGGLE:

<https://www.kaggle.com/datasets/msambare/fer2013>



Complete Machine Learning and Data Science: Zero to Mastery

4. <https://zerotomastery.io/courses/machine-learning-and-data-science-bootcamp/>



TensorFlow Developer Certificate in 2022: Zero to Mastery

5. <https://zerotomastery.io/courses/learn-tensorflow/>

6. <https://www.wikipedia.org/>

7. <https://mathworks.com/>

8. <https://stackoverflow.com/>