

# تقرير تفصيلي لمشروع تصنيف أمراض الجلد باستخدام Hybrid ResNet18

## 1. مقدمة

تُعد أمراض الجلد من الحالات الشائعة التي تتطلب تشخيصًا دقيقًا وسريعًا، خاصة في المراحل المبكرة من الإصابة. ومع التطور الكبير في تقنيات الرؤية الحاسوبية والذكاء الاصطناعي، أصبح من الممكن تطوير نماذج قادرة على تحليل صور الجلد وتصنيفها تلقائيًا بدقة عالية.

مع تطور تقنيات الذكاء الاصطناعي والرؤية الحاسوبية، أصبح من الممكن تطوير أنظمة ذكية تساعد الأطباء في تشخيص الحالات المرضية بسرعة ودقة. تُعد أمراض الجلد من الحالات الشائعة والمعقدة، حيث تتشابه مظاهرها الظاهرية، مما يجعل التشخيص اليدوي عرضة للخطأ أو التأخير.

في هذا السياق، يهدف هذا المشروع إلى تطوير نموذج يعتمد على تقنيات التعلم العميق لتصنيف تسعة أنواع من أمراض الجلد من خلال تحليل الصور الطبية. وقد تم اعتماد نسخة هجينة من الشبكة العصبية ResNet18 لتحقيق هذا الهدف بكفاءة.

لتنفيذ هذا المشروع بكفاءة واحترافية، تم الاعتماد على مجموعة من الأدوات والمكتبات البرمجية المتقدمة: تعمل هذه الأدوات معًا على تسريع عملية التطوير، وتحسين دقة التقييم، وتقديم نموذج قابل للتكامل مع الأنظمة الطبية المختلفة.

- **PyTorch**: مكتبة مفتوحة المصدر تُستخدم لبناء الشبكات العصبية وتدريبها، وتوفر مرونة كبيرة في تصميم النماذج.
- **Torchvision**: مكتبة مصاحبة لـ PyTorch تتضمن مجموعات بيانات جاهزة، ووظائف لتحميل الصور وتطبيق التحويلات عليها.
- **PyTorch Lightning**: إطار ميني على PyTorch ينظم عملية التدريب، يقلل من التكرار في الكود، ويدعم ميزات متقدمة مثل EarlyStopping و Logging والتصدير.
- **scikit-learn (sklearn)**: مكتبة تحليل إحصائي متخصصة في تقييم أداء النماذج، تم استخدامها لحساب دقة التوقع والاسترجاع ومتوسط F1.
- **matplotlib**: مكتبة لرسم المنحنيات البيانية التي تُستخدم لتحليل نتائج النموذج بصريًا، مثل منحنيات الخسارة والدقة.

## 2. البيانات وتحضيرها

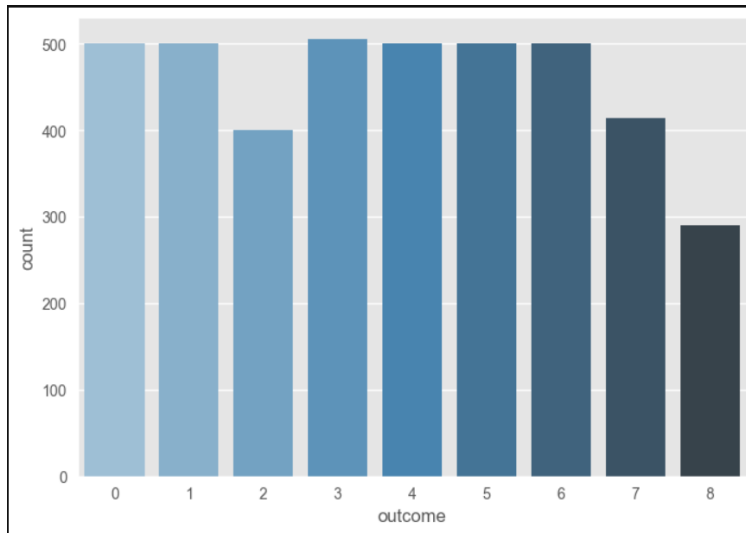
✓ مصدر البيانات:

تم استخدام الصور من مجلد مصنف (ImageFolder)، حيث تم تنظيم الصور داخل مجلدات فرعية تمثل كل منها فئة مختلفة من أمراض الجلد.

✓ عدد الفئات:

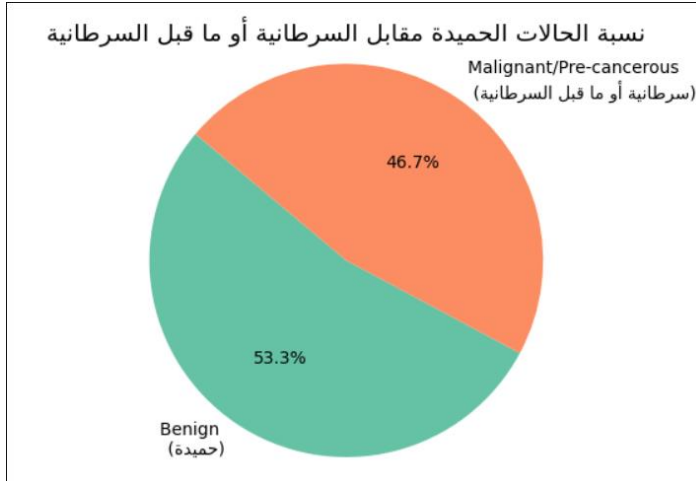
النموذج يقوم بتصنيف 9 أنواع مختلفة، وهي:

- Actinic Keratosis (500)
- Basal Cell Carcinoma (500)
- Dermatofibroma (400)
- Melanoma (505)
- Nevus (500)
- Pigmented Benign Keratosis (500)
- Seborrheic Keratosis (500)
- Squamous Cell Carcinoma (414)
- Vascular Lesion (290)



✓ التصنيف الطبي العام:

- حالات خبيثة أو ما قبل سرطانية (Malignant/Pre-cancerous): 1919 حالة
- حالات سليمة (Benign): 2190 حالة



### ✓التحويلات: (Transformations)

- تم استخدام مجموعة من التحويلات لزيادة تنوع البيانات وتحسين قدرة النموذج على التعميم:
  - تدوير عشوائي للصور بزاوية  $\pm 10$  درجات.
  - قلب أفقي عشوائي.
  - تغيير الحجم إلى 224 بكسل لأقصر ضلع.
  - اقتصاص مركزي لتوحيد الأبعاد.
  - تطبيع باستخدام القيم المستخدمة في نماذج ImageNet المتوسط والانحراف المعياري لكل قناة لونية.

### ✓تقسيم البيانات:

تم تقسيم البيانات إلى:

- 70% تدريب
- 15% تحقق (Validation)
- 15% اختبار (Testing)

### ✓معالجة عدم التوازن:

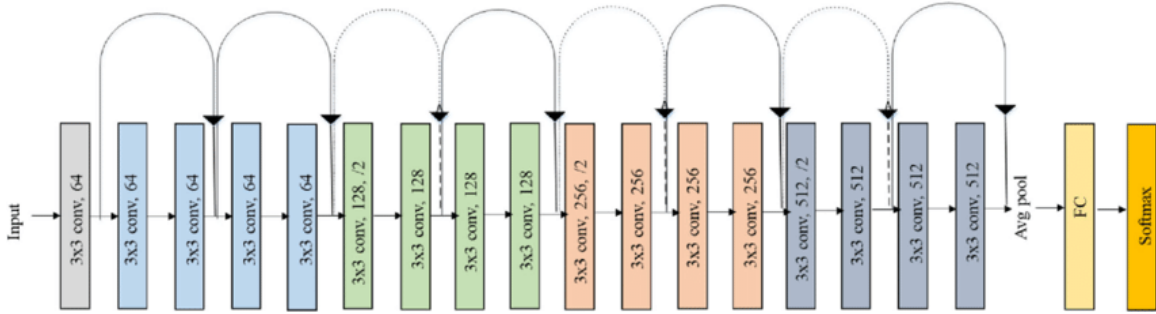
تم استخدام تقنية WeightedRandomSampler ، حيث يتم إعطاء وزن أكبر للفئات ذات التمثيل المنخفض في بيانات التدريب، لضمان توزيع عادل أثناء عملية التعلم. يتم حساب وزن كل فئة بشكل عكسي لحجمها النسبي، ويتم اختيار العينات أثناء التدريب بناءً على هذه الأوزان، مما يساعد النموذج على التعلم بعدالة من جميع الفئات.

### 3. تصميم النموذج Hybrid ResNet18 :

النسخة الهجينة من ResNet18 (Hybrid ResNet18) هي نموذج معدل يعتمد على ResNet18 المدرب مسبقاً، وقد تم تصميمه ليجمع بين مرونة التخصيص وكفاءة التدريب. وهي تمثل مزيجاً بين نموذج ResNet18 اليدوي باستخدام PyTorch التقليدي، والذي يعتمد على تجميد معظم الطبقات وتخصيص رأس التصنيف، ونموذج ResNet18 المبني باستخدام PyTorch Lightning المباشر، الذي يتم فيه تدريب النموذج بالكامل دون تخصيص كبير. في النسخة الهجينة، تم تجميد أغلب طبقات ResNet18 (باستثناء آخر 15 طبقة) للحفاظ على الميزات المستخلصة مسبقاً، مع استبدال الطبقة النهائية (fc) برأس مخصص يتكون من تسلسل  $\text{Linear} \rightarrow \text{ReLU} \rightarrow \text{Dropout} \rightarrow \text{Linear}$  وذلك لضبط النموذج ليتعامل مع تسع فئات مختلفة من أمراض الجلد.

أبرز ما يميز النسخة الهجينة:

- ✓ استخدام AdamW بمعدلات تعلم مختلفة لكل جزء (backbone) و (classifier) لتحقيق توازن في التحديثات.
- ✓ دمج WeightedRandomSampler في DataLoader للتعامل مع عدم توازن الفئات.
- ✓ تنظيم الكود والتدريب باستخدام PyTorch Lightning، مما أتاح استخدام تقنيات مثل EarlyStopping، والتسجيل الآلي، والتصدير إلى pt و onnx.



الشكل (1)

#### شكل (1): الهيكل الأصلي لنموذج ResNet-18

يوضح هذا الشكل بنية ResNet-18 التقليدية كما تم استخدامها في الأدبيات العلمية، والتي تتكون من سلسلة من الكتل المتبقية (Residual Blocks) يليها طبقة تصنيف نهائية (Fully Connected). في النسخة الهجينة المستخدمة في هذا المشروع، تم استخدام هذا الهيكل كـ **Backbone مع تجميد أغلب الطبقات** (باستثناء آخر 15 طبقة)، واستبدال الرأس (fc layer) برأس مخصص يتكون من:

$\text{Linear} \rightarrow \text{ReLU} \rightarrow \text{Dropout} \rightarrow \text{Linear}$

مما أتاح تحسين أداء النموذج وتقليل زمن التدريب، مع الحفاظ على القوة التمثيلية للطبقات المدربة مسبقاً.

هذه التعديلات مكنت النموذج من تحقيق دقة اختبار مرتفعة  $\approx$  (Test Accuracy 86%)، مع تقليل خطر الـ Overfitting، كما أظهر أداءً قوياً حتى مع البيانات الطبية غير المتوازنة، مما يجعل Hybrid ResNet18 هو الاختيار الأمثل في مثل هذا النوع من المهام التصنيفية المعقدة.

---

#### 4. التدريب (Training)

- تم استخدام PyTorch Lightning Trainer لإدارة التدريب بشكل منظم، حيث يوفر هذا الإطار أدوات متقدمة لتتبع الأداء، والتسجيل الآلي للمقاييس، وتبسيط تنفيذ الحلقات التدريبية.
  - تم تدريب النموذج على مدار 50 دورة (Epochs) كحد أقصى، مع مراقبة الأداء على مجموعة التحقق بعد كل دورة.
  - تم تفعيل ميزة EarlyStopping لإيقاف التدريب تلقائياً عند عدم تحسن دقة التحقق لفترة معينة، مما يقلل من خطر فرط التدريب ويوفر الوقت.
  - خلال عملية التدريب، تم تسجيل قيم الخسارة والدقة لكل من مجموعتي التدريب والتحقق، مما أتاح تحليل سلوك النموذج بدقة وتقييم مدى استقراره وتحسنه بمرور الوقت.
  - كما تم استخدام WeightedRandomSampler في DataLoader لضمان تمثيل عادل للفئات الأقل عدداً، مما حسن من استجابة النموذج للبيانات غير المتوازنة خلال التدريب.
- 

#### 5. التقييم والاختبار (Evaluation & Testing)

- تم تقييم النموذج باستخدام تقارير التصنيف (Classification Report) التي تتضمن دقة، استرجاع، ومتوسط F1 لكل فئة.
- كما تم استخدام مصفوفة الالتباس (Confusion Matrix) لتوضيح مدى التداخل بين الفئات المختلفة.

✓ نتائج التقييم:

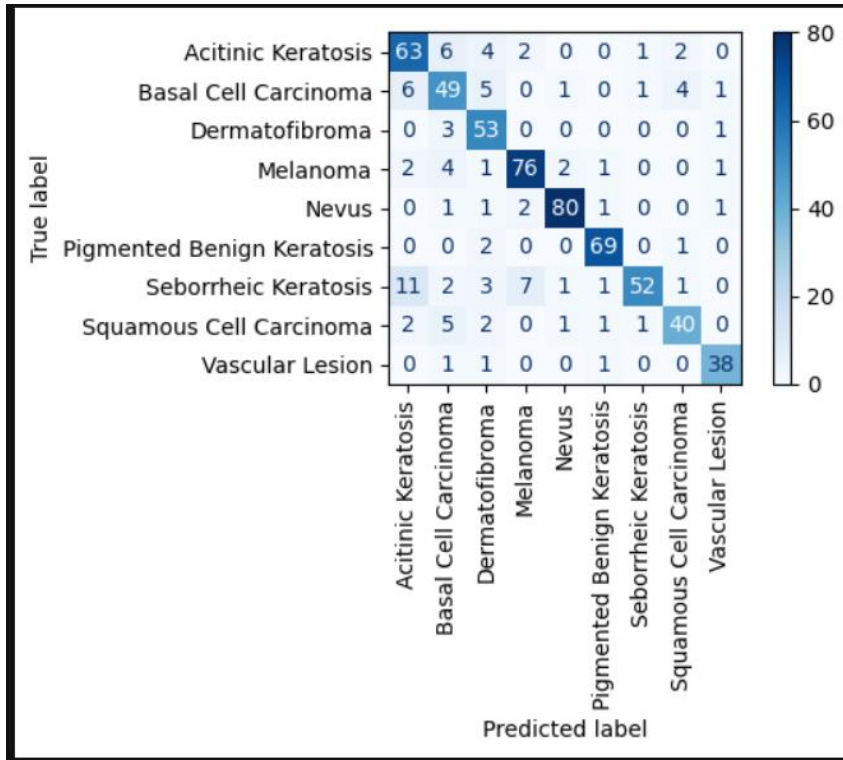
• **Test Accuracy: 85.92%**  
• **Test Loss: 0.459**

✓ الأداء لكل فئة:

- Acitinic Keratosis: F1-score = 0.778
- Basal Cell Carcinoma: F1-score = 0.710
- Dermatofibroma: F1-score = 0.822
- Melanoma: F1-score = 0.874
- Nevus: F1-score = 0.936
- Pigmented Benign Keratosis: F1-score = 0.945
- Seborrheic Keratosis: F1-score = 0.782

- Squamous Cell Carcinoma: F1-score = 0.800
- Vascular Lesion: F1-score = 0.916

- الدقة الكلية (accuracy): 84.14%
- المتوسط الحسابي (F1 (macro avg)): 0.840
- المتوسط المرجح (F1 (weighted avg)): 0.841



## 6. منحنيات الأداء (Training Curves)

تم رسم منحنيات الخسارة والدقة باستخدام matplotlib

- منحنى يوضح انخفاض الخسارة عبر الدورات.
- منحنى يوضح تحسن الدقة على كل من مجموعة التدريب والتحقق.

يساعد هذا التحليل البصري على اكتشاف أي علامات فرط تدريب (Overfitting) أو ضعف تعلم.

## 7. تصدير النموذج (Exporting the Model)

لتسهيل دمج النموذج في تطبيقات مختلفة، تم تصديره بصيغتين:

- **TorchScript (.pt):** للتشغيل على أجهزة الموبايل.
- **ONNX (.onnx):** للدمج مع أنظمة أخرى مثل ONNX Runtime أو تطبيقات Web/Embedded.

## 8. الخاتمة

يمثل هذا المشروع خطوة فعالة نحو بناء نظام ذكي لتصنيف أمراض الجلد بدقة عالية باستخدام نموذج معدل قائم على ResNet18. استطاع النموذج تحقيق نتائج قوية من حيث الدقة والانحياز المنخفض، كما أثبت كفاءته في التعامل مع بيانات غير متوازنة من خلال استخدام أدوات مثل WeightedRandomSampler.

تميز المشروع بتوظيف إطار PyTorch Lightning الذي ساعد على تنظيم الكود، تسهيل التدريب، وتطبيق تقنيات مثل EarlyStopping، مما زاد من كفاءة النموذج وسرعة تطويره.

من أبرز نقاط القوة التي ظهرت أثناء التنفيذ:

- كفاءة النموذج في التعميم على فئات متعددة.
- جودة النتائج رغم بساطة التعديل على النموذج الأساسي.
- سهولة التصدير والاستخدام في تطبيقات مختلفة.

ولضمان التطوير المستقبلي للنموذج، يمكن التفكير في ما يلي:

- تجربة نماذج أعمق مثل ResNet50 أو EfficientNet.
- تطبيق تقنيات Augmentation أكثر تقدماً.
- استخدام مجموعات بيانات أكبر وأكثر تنوعاً.
- دمج النموذج في تطبيق واقعي مثل تطبيق موبايل للطبيب العام.

بذلك يُعد هذا المشروع نواة قوية لإنتاج حلول ذكية يمكن أن تُستخدم في دعم القرار الطبي، وتسريع عمليات التشخيص، ورفع مستوى الدقة في الكشف المبكر عن أمراض الجلد.