



الجمهورية العربية السورية  
المعهد العالي للعلوم التطبيقية والتكنولوجيا  
قسم النظم المعلوماتية  
اختصاص هندسة البرمجيات والذكاء الصناعي  
٢٠٢٤ - ٢٠٢٥

## تقدير العمق أحادي الرؤية باستخدام تقنيات تقطير المعرفة

إعداد  
ميّاس الخطيب

إشراف  
د. عمر حمدون

24/07/2025

# الخلاصة

يعالج هذا المشروع تحدي تشغيل نماذج تقدير العمق أحادي الرؤية، التي تتطلب قدرات حاسوبية فائقة، على الهواتف المحمولة محدودة الموارد. تم اعتماد تقنية تقطير المعرفة (Knowledge Distillation) لتطوير نموذج فعال يوازن بين الدقة والأداء في الزمن الحقيقي. تعتمد المنهجية على نقل المعرفة من نموذج "معلم" متطور وهو Depth Anything V2، إلى نموذج "طالب" خفيف الوزن تم تصميمه وتدريبه باستخدام إطار عمل PyTorch. تتميز بنية النموذج الطالب المبتكرة باستخدام المرمز MobileViT-XS، الذي يجمع بين الكفاءة والقدرة على فهم السياق الشمولي، مع مفكك ترميز مخصص وخفيف الوزن (MiniDPT). تم توجيه عملية التدريب عبر دالة خسارة مركبة تضمن نقل المعرفة بفعالية من خلال مطابقة خرائط العمق والتدرجات والسمات الداخلية. علاوة على ذلك، يتضمن المشروع بناء تطبيق للهواتف المحمولة باستخدام Flutter لإثبات قدرات النموذج في الزمن الحقيقي. ينجح هذا النهج في تحقيق توازن مثالي بين دقة التقدير ومتطلبات الأداء على الأجهزة المحمولة، مما يفتح آفاقاً لتطبيقات واسعة في الواقع المعزز والمساعدة في التنقل.

## Abstract

This project addresses the challenge of running computationally intensive monocular depth estimation models on mobile devices. We employ knowledge distillation to create an efficient model that balances accuracy with real-time performance on resource-limited hardware. The methodology involves transferring knowledge from a state-of-the-art teacher model, Depth Anything V2, to a lightweight student model designed and trained using the PyTorch framework. The novel student architecture features a MobileViT-XS encoder, chosen for its ability to capture global context efficiently, and a custom, lightweight MiniDPT decoder. Training is guided by a composite loss function that includes scale-invariant, gradient, and feature-matching losses to ensure robust knowledge transfer. The project culminates in a Flutter-based mobile application that successfully demonstrates the model's real-time capabilities. This approach achieves an optimal balance between performance and accuracy, paving the way for practical applications in augmented reality and navigation assistance.

# المحتويات

i	الخلاصة-----
ii	المحتويات-----
iv	قائمة الأشكال-----
v	قائمة الجداول-----
vi	الاختصارات-----
1	الفصل الأول: التعريف بالمشروع-----
1	1.1- مقدمة.....
1	2.1- الهدف من المشروع.....
1	3.1- المتطلبات.....
1	1.3.1- المتطلبات الوظيفية.....
2	2.3.1- المتطلبات غير الوظيفية.....
3	الفصل الثاني: دراسة مرجعية-----
3	1.2- مقدمة.....
3	2.2- الأساليب التقليدية.....
5	3.2- أساليب تعلم الآلة.....
5	1.3.2- التعلم الخاضع للإشراف.....
5	2.3.2- التعلم غير الخاضع للإشراف والخاضع للإشراف الذاتي.....
5	4.2- معايير التقييم.....
6	الفصل الثالث: المنهجية المتبعة-----
6	1.3- التقنيات المستخدمة.....
6	2.3- تقنية تقطير المعرفة.....
6	1.2.3- آلية عمل تقطير المعرفة.....
6	2.2.3- أنواع المعرفة القابلة للتقطير.....
7	3.3- منهجية التدريب.....
8	4.3- تصميم النموذج الطالب.....
8	1.4.3- بنية النموذج المعلم.....
9	2.4.3- بنية النموذج الطالب.....
11	الفصل الرابع: التنفيذ والتطبيق-----
11	1.4- البيئات والأدوات المستخدمة.....
11	2.4- بناء النموذج وتدريبه.....

11	1.2.4- تهيئة البيئة.....
11	2.2.4- هيكل الملفات.....
13	3.2.4- محاولات التدريب.....
14	3.4- بناء تطبيق الموبايل.....
14	1.3.4- البنية الهيكلية.....
15	2.3.4- تحسينات الأداء.....
15	3.3.4- استعراض وظائف التطبيق.....
17	الفصل الخامس: النتائج والمناقشة-----
17	1.5- بيئة الاختبار.....
17	2.5- التقييم الكمي.....
17	3.5- التقييم النوعي.....
18	4.5- قياسات الأداء.....
18	5.5- المناقشة.....
18	6.5- آفاق مستقبلية.....
20	المراجع-----

## قائمة الأشكال

- الشكل 1- طرائق تقدير العمق.....3
- الشكل 2- الرؤية المجسمة.....3
- الشكل 3- بنية من الحركة.....4
- الشكل 4- العمق من التركيز/عدم التركيز.....4
- الشكل 5- الخسارة اللوغارتمية المستقلة عن المقياس.....7
- الشكل 6- خسارة مطابقة التدرج.....8
- الشكل 7- بنية النموذج المعلم.....9
- الشكل 8- بنية الطالب.....10
- الشكل 9- نتائج محاولة التدريب الأولى.....13
- الشكل 10- نتائج محاولة التدريب الثانية.....13
- الشكل 11- نتائج محاولة التدريب الثالثة.....14
- الشكل 12- اختيار صورة وتقدير العمق وعرض النتائج الشكل 13- تخصيص العرض وحفظ الصورة والنمط الليلي.....15
- الشكل 14- مقارنة بصرية.....17

## قائمة الجداول

- الجدول 1- مقارنة بين الطرائق التقليدية وأساليب تعلم الآلة لتقدير العمق.....5
- الجدول 2- أوزان الترجيح لمكونات تابع الخسارة.....14
- الجدول 3- التقييم الكمي للنموذج الطالب.....17
- الجدول 4- قياس الأداء للنموذج.....18

# الاختصارات

# الفصل الأول: التعريف بالمشروع

## 1.1- مقدمة

يعد تقدير العمق من صورة واحدة مهمة محورية في مجال الرؤية الحاسوبية، حيث يفتح الباب أمام تطبيقات ثورية تمتد من القيادة الذاتية والملاحة الروبوتية إلى الواقع المعزز وإعادة بناء المشاهد ثلاثية الأبعاد. في السنوات الأخيرة، حققت نماذج التعلم العميق الضخمة، مثل DPT و MiDas، دقة مذهلة في هذه المهمة، لكن أدائها المتميز يأتي على حساب متطلبات حسابية هائلة، مما يجعلها غير عملية للتطبيقات التي تعمل على أجهزة محدودة الموارد كالهواتف المحمولة. يعالج هذا المشروع هذا التحدي الجوهرى: **كيف يمكننا سد الفجوة بين دقة النماذج الكبيرة ومتطلبات الأداء والكفاءة للأجهزة المحمولة؟** الحل يكمن في تقنية تقطير المعرفة (**Knowledge Distillation**). من خلال هذه التقنية، نقوم بتدريب نموذج "طالب" خفيف الوزن وفعال لتقليد سلوك ومخرجات نموذج "معلم" ضخيم وعالي الأداء. يهدف هذا المشروع إلى تصميم وتدريب نموذج طالب مخصص، ونقل المعرفة إليه من نموذج **Depth Anything V2** المتطور، وأخيراً، إثبات جدوى هذا النهج من خلال بناء تطبيق عملي يعرض النتائج في الزمن الحقيقي.

## 2.1- الهدف من المشروع

الهدف الرئيسى من هذا المشروع هو إثبات أن تقنية تقطير المعرفة هي استراتيجية فعالة لتطوير نماذج تقدير عمق عالية الجودة وقابلة للتشغيل على الأجهزة المحمولة. لتحقيق ذلك، تم تحديد الأهداف الفرعية التالية:

1. **تصميم وتنفيذ نموذج طالب (Student Model):** بناء معمارية شبكة عصبونية خفيفة الوزن وسريعة حسابياً، مصممة خصيصاً لتعمل بكفاءة على الأجهزة محدودة الموارد.
2. **استخدام نموذج معلم (Teacher Model) متطور:** الاستفادة من نموذج **Depth Anything V2** المدرب مسبقاً كـ "معلم" يمتلك خبرة واسعة في فهم المشاهد وتقدير العمق بدقة.
3. **تطبيق استراتيجية تقطير معرفة شاملة:** تدريب النموذج الطالب لتقليد مخرجات المعلم على مستويات متعددة، بما في ذلك خريطة العمق النهائية، تدرجات الصورة، والتمثيلات الداخلية (feature maps)، لضمان نقل أكبر قدر ممكن من "المعرفة".
4. **تقييم الأداء والدقة بشكل صارم:** قياس أداء النموذج الطالب من حيث الدقة (باستخدام مقاييس مثل Abs Rel و RMSE على مجموعة بيانات KITTI) ومقارنتها بأداء المعلم، بالإضافة إلى قياس كفاءته الحاسوبية (زمن الاستدلال، حجم النموذج).
5. **بناء عرض توضيحي عملي (Proof of Concept):** تطوير تطبيق للهواتف المحمولة باستخدام Flutter لدمج النموذج الطالب المُقَطَّر وعرض قدرته على تقدير العمق في الزمن الحقيقي، لإثبات قابليته للاستخدام في تطبيقات العالم الحقيقي.

## 3.1- المتطلبات

لضمان تحقيق الهدف من هذا المشروع -وهو بناء عرض توضيحي (Proof of Concept) يثبت فعالية النموذج المقترح على الأجهزة محدودة الموارد- تم تحديد مجموعة من المتطلبات الأساسية. تركز هذه المتطلبات على الوظائف الدنيا اللازمة لعرض قدرات النموذج وقياس أدائه في بيئة تشغيل حقيقية، ولا تهدف إلى بناء منتج تجاري متكامل.

### 1.3.1- المتطلبات الوظيفية

1. **إدخال الصورة:**
  - يجب أن يتمكن المستخدم من التقاط صورة جديدة باستخدام كاميرا الجهاز.
  - يجب أن يتمكن المستخدم من اختيار صورة موجودة مسبقاً من معرض الصور في الجهاز.
2. **معالجة الصورة:**
  - يجب أن يقوم النظام بتمرير الصورة المدخلة إلى نموذج الذكاء الاصطناعي المبني لتقدير العمق.
  - يجب أن يقوم النظام بتوليد خريطة عمق (Depth Map) ملونة تمثل المسافات في الصورة الأصلية.



### 3. عرض النتائج:

- يجب أن يعرض التطبيق الصورة الأصلية التي أدخلها المستخدم.
- يجب أن يعرض التطبيق خريطة العمق الناتجة بجانب الصورة الأصلية للمقارنة.
- يجب أن يعرض التطبيق الزمن اللازم لتوليد خريطة العمق مقاساً بالملي ثانية.

### 4. إدارة المخرجات:

- يجب أن يتمكن المستخدم من حفظ خريطة العمق الناتجة (أو الصورة الأصلية مع خريطتها) في ذاكرة الجهاز.

## 2.3.1- المتطلبات غير الوظيفية

### 1. الأداء:

- زمن الاستجابة: يجب ألا يتجاوز زمن عملية تقدير العمق (زمن الاستدلال) الـ **200 ms** على الأجهزة المحمولة متوسطة المواصفات.
- استهلاك الذاكرة: يجب ألا يتجاوز استهلاك التطبيق للذاكرة العشوائية (RAM) حداً معيناً (مثلاً 100 ميجابايت) أثناء التشغيل لمنع الانهيار.
- حجم التطبيق: يجب أن يكون حجم ملف التثبيت (APK/IPA) صغيراً قدر الإمكان (يفضل ألا يتجاوز 50 ميجابايت) لتسهيل التنزيل والتثبيت.

### 2. الدقة:

- يجب أن يحقق النموذج الطالب دقة مقبولة على مجموعة بيانات الاختبار (مثل KITTI)، مع تحقيق توازن بين الدقة والأداء. على سبيل المثال، يجب أن يهدف النموذج لتحقيق خطأ نسبي مطلق (Abs Rel) أقل من 0.1.

### 3. قابلية الاستخدام:

- يجب أن تكون واجهة المستخدم (UI) بسيطة وبديهية، ولا تتطلب من المستخدم معرفة تقنية مسبقة.
- يجب أن يقدم التطبيق إرشادات واضحة وردود فعل للمستخدم (مثل إظهار مؤشر "جاري المعالجة...").

### 4. التوافقية:

- يجب أن يعمل التطبيق بشكل صحيح على كل من نظامي التشغيل **iOS** و **Android**.
- يجب أن يدعم التطبيق إصدارات محددة من أنظمة التشغيل (Android 8.0 فما فوق، و iOS 12 فما فوق).

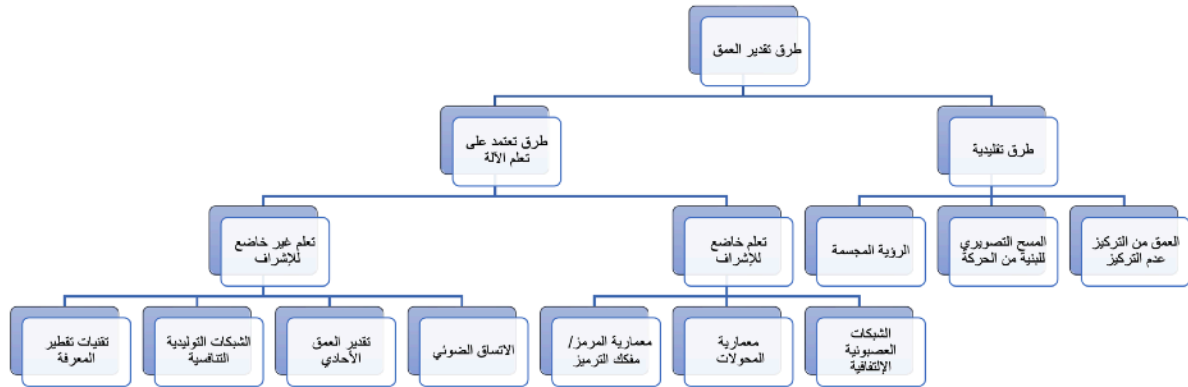
### 5. الوثوقية:

- يجب أن يكون التطبيق مستقرًا، مع معدل انهيار منخفض جدًا أثناء الاستخدام العادي.
- يجب أن يتعامل التطبيق مع الأخطاء بشكل سليم (مثل عدم القدرة على تحميل الصورة أو انقطاع الكاميرا) دون أن ينهار.

## الفصل الثاني: دراسة مرجعية

### 1.2- مقدمة

يتضمن تقدير العمق عملية التنبؤ بمسافة الأجسام في مشهد ما من منظور كاميرا محدد. قبل ظهور تعلم الآلة، اعتمد هذا المجال بشكل أساسي على تقنيات هندسية وضوئية. نستعرض في هذا الفصل الطرائق التقليدية والحديثة المعتمدة على تعلم الآلة.

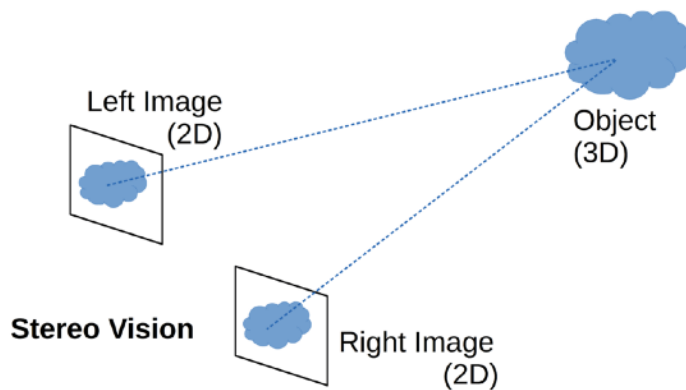


الشكل 1- طرائق تقدير العمق

### 2.2- الأساليب التقليدية

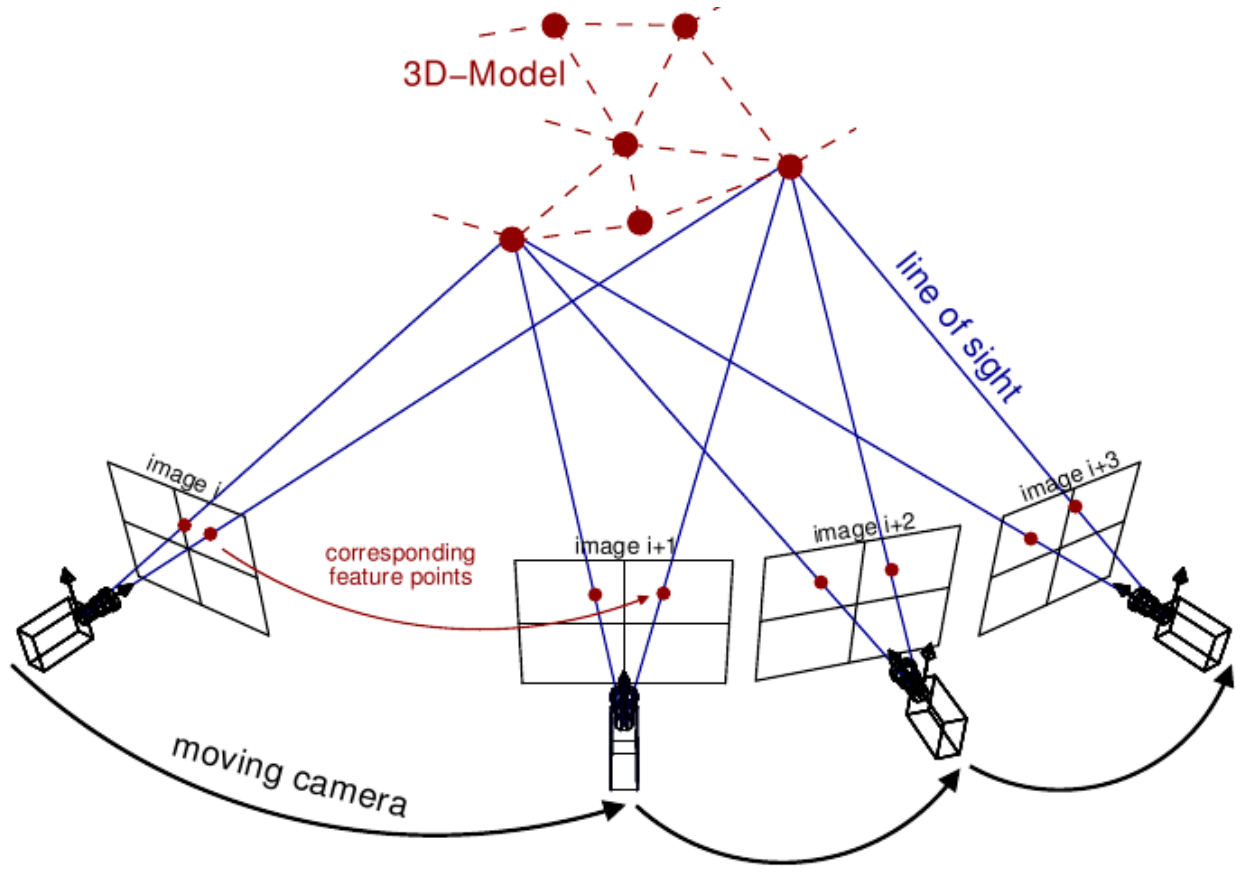
اعتمدت الأساليب التقليدية على مبادئ هندسية لاستنتاج العمق، ومن أهمها:

- **الرؤية المجسمة (Stereo Vision):** تستخدم كاميرتين أو أكثر لالتقاط صور من وجهات نظر مختلفة قليلاً ومن ثم حساب خريطة العمق عبر التثليث. تتطلب هذه الطريقة معايير دقيقة وتواجه صعوبات في المناطق عديمة اللمس.



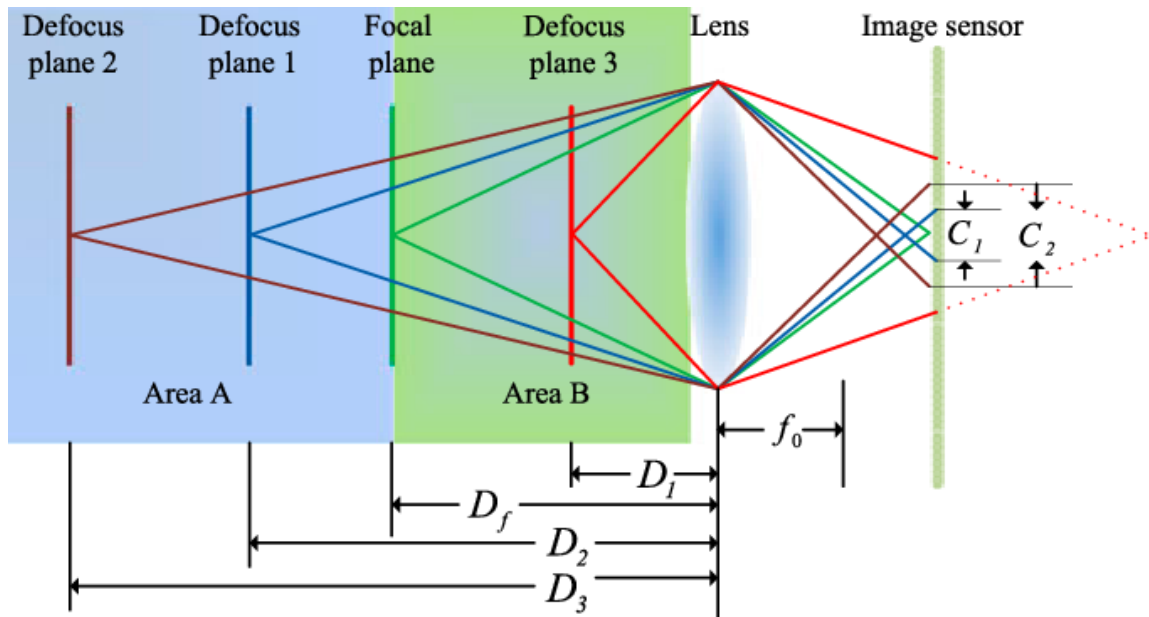
الشكل 2- الرؤية المجسمة

- **بنية من الحركة (Structure from Motion - SfM):** تقدر العمق عبر تحليل حركة نقاط مميزة عبر سلسلة من الصور، مما يسمح بإعادة بناء هيكل المشهد ثلاثي الأبعاد. هذه الطريقة حساسة لأخطاء التتبع ومكلفة حسابياً.



الشكل 3- بنية من الحركة

• **العمق من التركيز/عدم التركيز (Depth from Focus/Defocus):** تقدر العمق بناءً على درجة الضبابية (الغشاوة) في الصورة عند أطوال بؤرية مختلفة. ومع ذلك تتطلب هذه الطريقة ظروف إضاءة قابلة للتحكم وتقتصر على مجال عمق الكاميرا.



الشكل 4- العمق من التركيز/عدم التركيز

## 3.2- أساليب تعلم الآلة

أحدثت نماذج تعلم الآلة، والتعلم العميق خصوصاً، نقلة نوعية في المجال. يمكن تقسيمها إلى:

### 1.3.2- التعلم الخاضع للإشراف

تتطلب أساليب التعلم الخاضع للإشراف مجموعات بيانات موسومة بخرائط عمق مرجعية. حققت هذه الأساليب أداءً متقدماً من خلال الاستفادة من مجموعات البيانات واسعة النطاق وبنى الشبكات العصبية المتقدمة. من أبرز معماريات هذا النهج:

- **الشبكات العصبونية الالتفافية (CNNs):** قدم أحد أول نماذج التعلم العميق لتقدير العمق، باستخدام شبكة CNN متعددة المقاييس للتنبؤ بالعمق من صورة واحدة. أظهر نهجهم إمكانات التعلم العميق لهذه المهمة.
- **معمارية المرمز-مفك الترميز:** تُستخدم شبكات U-Net والمشابهاة على نطاق واسع للتنبؤ الكثيف بالعمق. تجمع هذه البنى بين الميزات الدلالية عالية المستوى والتفاصيل المكانية منخفضة المستوى لإنتاج خرائط عمق دقيقة.
- **نماذج المحولات:** ناقشت الأعمال الحديثة استخدام محولات الرؤية (ViTs) في تطبيقات تقدير العمق. تلتقط المحولات السياق الشمولي بشكل أكثر فعالية من CNNs، مما يؤدي إلى تحسين الأداء في المشاهد المعقدة.

### 2.3.2- التعلم غير الخاضع للإشراف والخاضع للإشراف الذاتي

لا تتطلب الأساليب غير الخاضعة للإشراف وشبه الخاضعة للإشراف (إشراف ذاتي) خرائط عمق مرجعية، مما يجعلها أكثر قابلية للتوسع وعملية للتطبيقات الواقعية. تعتمد على مفاهيم مثل:

- **الاتساق الضوئي:** يعتمد على تقليل الخطأ الضوئي بين الصور المتوقعة والأصلية باستخدام العمق المتوقع ووضع الكاميرا. يستغل هذا النهج القيود الهندسية لتدريب النموذج دون بيانات موسومة.
- **الشبكات التوليدية التنافسية (GANs):** تم استخدام GANs لتوليد خرائط عمق واقعية من خلال تعلم توزيع البيانات. يمكن لهذه النماذج تحسين جودة تنبؤات العمق من خلال دمج التدريب التنافسي.
- **تقنيات التقطير:** حيث يتم الاعتماد على نموذج مُدرَّب مسبقاً لينتج خرائط عمق شبه موثوقة ويتم تدريب النموذج المرجو عليها. ويكون النموذج المدرب اصغر من النموذج المدرب.

المعيار	الطرائق التقليدية	تعلم الآلة
الدقة	جيدة	عالية (مع التدريب الكافي)
احتياج البيانات	لا حاجة	كمية ضخمة موسومة وغير موسومة (لتحسين قابلية التعميم)
السرعة	بطيئة (وقد تحقق سرعة تصلح للزمن الحقيقي ولكن في حال الرؤية المجسمة)	عالية

الجدول 1- مقارنة بين الطرائق التقليدية وأساليب تعلم الآلة لتقدير العمق

## 4.2- معايير التقييم

يتم تقييم نماذج تقدير العمق باستخدام مجموعة متنوعة من المقاييس، بما في ذلك:

- **الخطأ النسبي المطلق (Absolute Relative Difference):** يقيس الفرق النسبي بين العمق المتوقع والعمق المرجعي.
- **الخطأ النسبي التربيعي (Squared Relative Difference):** يقيس مربع الفرق النسبي بين العمق المتوقع والعمق المرجعي، يؤكد على الأخطاء الكبيرة من خلال تربيع الفروق.
- **جذر متوسط الخطأ التربيعي (Root Mean Squared Error):** يحسب الجذر التربيعي لمتوسط الفروق التربيعية.
- **الخطأ اللوغاريتمي ثابت القياس (Scale Invariant Log Loss):** يأخذ في الاعتبار غموض القياس في تقدير العمق الأحادي. حيث لا نعرف العمق الفعلي بالتر ولكن نعرف العمق النسبي بين الأغراض في الصورة.
- **عتبات الدقة (Accuracy Threshold):** نسبة البكسلات التي يكون فيها العمق المتوقع ضمن عتبة معينة من العمق المرجعي (مثل،  $1.25 < \delta$ ).

## الفصل الثالث: المنهجية المتبعة

### 1.3- التقنيات المستخدمة

1. **تقنية تقطير المعرفة Knowledge Distillation:** تُعد تقنية تقطير المعرفة الاستراتيجية المحورية التي يقوم عليها هذا المشروع، وهي تهدف إلى حل التناقض بين دقة النماذج الضخمة والمتطلبات الحاسوبية للأجهزة محدودة الموارد مثل الهواتف المحمولة. الفكرة الأساسية هي تدريب نموذج صغير وفعال، يُطلق عليه اسم "النموذج الطالب" (Student Model)، ليحاكي سلوك ومخرجات نموذج ضخم وعالي الأداء، يُعرف باسم "النموذج المعلم" (Teacher Model).
2. **بنية المرمز - مفك الترميز (Encoder-Decoder Architecture):** هو تصميم شائع للشبكات العصبونية يُستخدم لمهام مثل تقدير العمق. يقوم بمعالجة الصورة لاستخلاص الميزات (المرمز) ثم يستخدم هذه الميزات لتوليد خريطة العمق (مفك الترميز).
3. **محولات الرؤية (Vision Transformers - ViT):** هي بنية شبكة عصبونية فعالة بشكل خاص في التقاط السياق العام للصورة. يعتمد النموذج المعلم، Depth Anything V2، على هذه التقنية.

### 2.3- تقنية تقطير المعرفة

**تقطير المعرفة** هي إحدى تقنيات ضغط النماذج (Model Compression) المتقدمة في مجال تعلم الآلة، والتي تهدف إلى نقل "المعرفة" من نموذج كبير ومعقد وعالي الأداء، يُعرف باسم **النموذج المعلم (Teacher Model)**، إلى نموذج أصغر حجمًا وأكثر كفاءة من الناحية الحسابية، يُطلق عليه **النموذج الطالب (Student Model)**. الفكرة مستوحاة من العلاقة بين المعلم والطالب في العالم الحقيقي، حيث لا يكتفي المعلم بإعطاء الإجابة النهائية للطالب، بل يشرح له طريقة التفكير والمنطق الذي يؤدي إلى تلك الإجابة.

#### 1.2.3- آلية عمل تقطير المعرفة

تعتمد الآلية على تدريب النموذج الطالب ليس فقط على البيانات الأصلية المصنفة (Ground Truth)، بل أيضًا على مخرجات النموذج المعلم. تكمن قوة هذه التقنية في أن النموذج المعلم يقدم إشارات تدريبية أكثر ثراءً من مجرد الإجابة الصحيحة.

1. **الأهداف الصلبة (Hard Targets):** هي الإجابات الصحيحة الموجودة في مجموعة البيانات (على سبيل المثال: "هذه الصورة تحتوي على قطة"). التدريب باستخدام هذه الأهداف فقط قد يكون صعبًا على نموذج صغير.
2. **الأهداف الناعمة (Soft Targets):** هي المخرجات الاحتمالية الكاملة للنموذج المعلم. على سبيل المثال، في مسألة تصنيف بسيطة، بدلاً من أن يقول المعلم "هذه قطة" بنسبة 100%، قد يقول "هذه قطة بنسبة 90%، كلب بنسبة 5%، أرنب بنسبة 3%، إلخ". هذه التوزيع الاحتمالية، التي تُعرف بـ "**المعرفة المظلمة (Dark Knowledge)**"، تحتوي على معلومات ضمنية غنية حول مدى تشابه الفئات المختلفة من منظور المعلم. عندما يتعلم الطالب محاكاة هذه "الأهداف الناعمة"، فإنه يتعلم بنية العلاقات بين الفئات، مما يؤدي إلى تعميم أفضل وأداء أعلى.

#### 2.2.3- أنواع المعرفة القابلة للتقطير

- لا تقتصر عملية التقطير على مخرجات النموذج النهائية فقط، بل يمكن نقل أنواع متعددة من المعرفة لتعزيز أداء الطالب:
- **التقطير القائم على الاستجابة (Response-Based Distillation):** وهو النوع الكلاسيكي الذي يعتمد على محاكاة "الأهداف الناعمة" من الطبقة الأخيرة للمعلم.
  - **التقطير القائم على الميزات (Feature-Based Distillation):** في هذا النوع، يتم تدريب الطالب على محاكاة خرائط الميزات (Feature Maps) من الطبقات الوسطى للمعلم. هذا يجبر الطالب على تعلم نفس التسلسل الهرمي لاستخلاص الميزات الذي يستخدمه المعلم، مما يساعده على فهم البيانات بشكل أعمق.
  - **التقطير القائم على العلاقات (Relation-Based Distillation):** يركز هذا النوع على نقل العلاقات بين نقاط البيانات أو طبقات الميزات، مثل محاكاة خرائط الانتباه (Attention Maps) أو مصفوفات الارتباط بين الميزات.

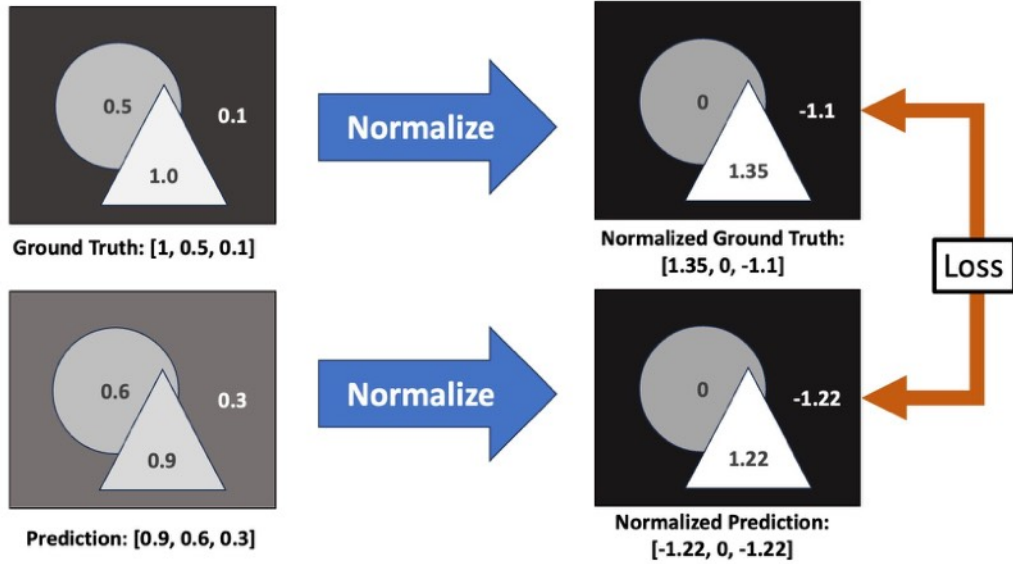
### 3.3- منهجية التدريب

تعتمد منهجية التدريب على إطار عمل المعلم والطالب (Teacher-Student) لتقطير وضغط المعرفة (الخبرة) من النموذج العملاق للنموذج الأصغر.

- النموذج المعلم (Teacher Model): سيتم استخدام نموذج **Depth Anything V2** (النسخة Large أو Small) كمعلم. هذا النموذج مدرب مسبقاً على كم هائل من البيانات ويقدم نتائج دقيقة جداً ولكنه غير مناسب للعمل على الهواتف المحمولة بسبب حجمه الكبير ومتطلباته الحاسوبية.
- النموذج الطالب (Student Model): سيتم تصميم نموذج جديد بمعمارية خفيفة (مثل MobileNet، MobileViT، أو معمارية مخصصة) ليكون فعالاً من حيث عدد البارامترات وسرعة الاستدلال.

تتم عملية التقطير عبر تدريب النموذج الطالب على تصغير دالة خسارة (Loss Function) مركبة. تتكون دالة الخسارة في حالتنا من أربعة مكونات رئيسية:

1. **خسارة العمق (Scale Invariant Logarithmic Loss - SILog):** هي المكون الأساسي الذي يقيس دقة خريطة العمق المتوقعة من الطالب (d) مقارنة بتلك الصادرة عن المعلم (d\*). تستخدم هذه الخسارة مقياساً لوغاريتمياً ثابتاً تجاه التكبير (Scale-Invariant) لضمان الدقة الهيكلية.

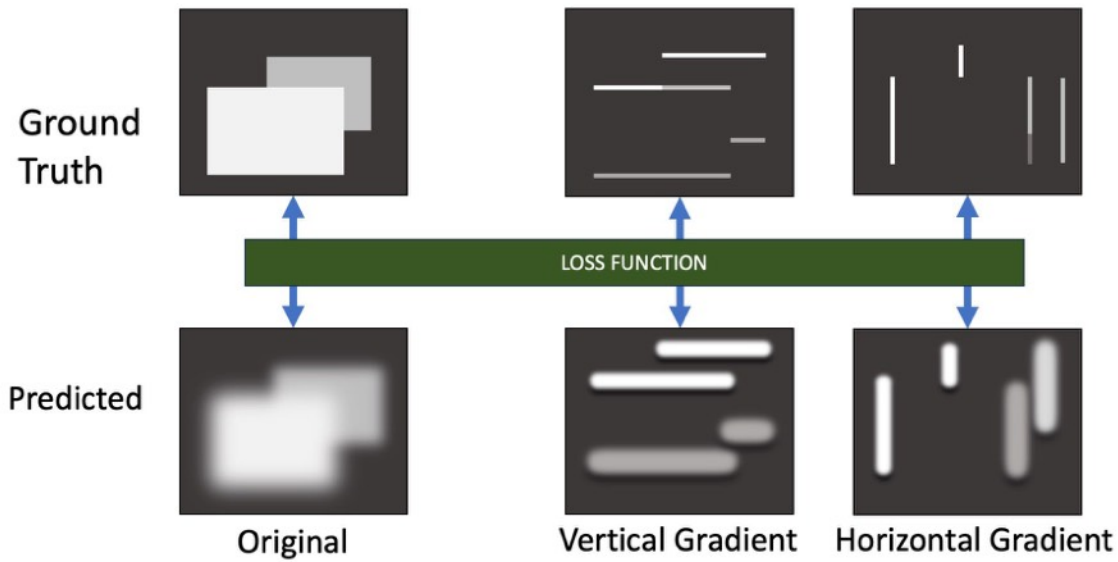


الشكل 5- الخسارة اللوغارتمية المستقلة عن المقياس

$$L_{silog} = \frac{1}{N} \sum_i (\log(d_i) - \log(d_i^*))^2 - \frac{1}{N^2} \left( \sum_i (\log(d_i) - \log(d_i^*)) \right)^2$$

حيث N هو عدد البكسلات في الصورة، و  $d_i$  يمثل كل بكسل. هذه الخسارة تضمن أن يكون هيكل العمق صحيحاً بغض النظر عن المقياس (scale).

2. **خسارة مطابقة التدرج (Gradient Matching Loss):** تجبر هذه الخسارة تدرجات (gradients) خريطة العمق للطالب على أن تكون مشابهة لتدرجات خريطة العمق للمعلم. هذا الأمر حاسم للحفاظ على الحواف والتفاصيل الدقيقة في المشهد.



الشكل 6- خسارة مطابقة التدرج

$$L_{grad} = \frac{1}{N} \sum_i (| \nabla_x(d_i) - \nabla_x(d_i^*) | + | \nabla_y(d_i) - \nabla_y(d_i^*) |)$$

حيث  $\nabla_x$  و  $\nabla_y$  تمثلان معاملات التدرج (مثل Sobel filter) في الاتجاهين الأفقي والعمودي، وتمثل  $N$  عدد البكسلات في الصورة.

3. **خسارة مطابقة خرائط الميزات (Feature Matching Loss):** بدلاً من تقليد المخرجات النهائية فقط، تهدف هذه الخسارة إلى جعل الخرائط الميزة في الطبقات الوسطى للطالب ( $f$ ) مشابهة لتلك الموجودة في المعلم ( $f^*$ ). هذا يساعد الطالب على تعلم "التمثيلات الداخلية" التي يستخدمها المعلم لفهم المشهد.

$$L_{feat} = \frac{1}{M} \sum_i |f_i - f_i^*|$$

حيث  $M$  تمثل عدد الميزات في خرائط الميزات.

4. **خسارة مناطق الانتباه (Attention Matching Loss):** تقوم هذه الخسارة بحساب "خريطة انتباه" مبسطة لكل من الطالب ( $attn$ ) والمعلم ( $attn^*$ ) من خلال تلخيص الخرائط الميزة. من خلال مطابقة هذه الخرائط، يتم تشجيع الطالب على التركيز على نفس المناطق المكانية الهامة في الصورة التي يركز عليها المعلم.

$$L_{attn} = \frac{1}{M'} \sum_i (attn - attn^*)^2$$

يتم تجميع هذه الخسائر في دالة خسارة كلية واحدة باستخدام معاملات ترجيح ( $\lambda$ ) لكل مكون:

$$L_{total} = \lambda_{silog} L_{silog} + \lambda_{grad} L_{grad} + \lambda_{feat} L_{featt} + \lambda_{attn} L_{attn}$$

سيتم ضبط معاملات الترجيح أثناء التنفيذ العملي بما يتناسب مع نتائج الاختبارات.

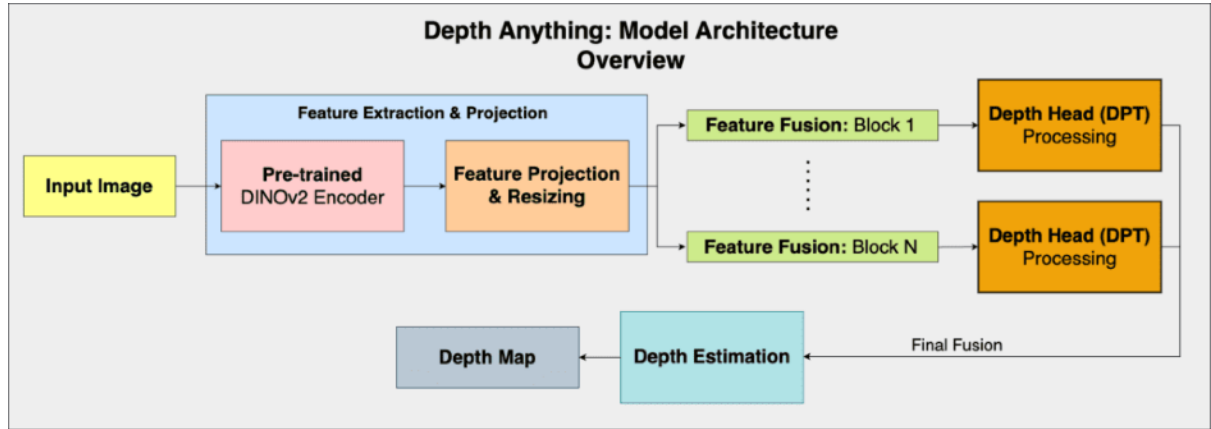
### 4.3- تصميم النموذج الطالب

تم تصميم واختبار عدة نماذج عبر تغيير المرمز أو تغيير بنية مفكك الترميز، ولكن سنذكر فقط النموذج المعتمد آخرًا.

#### 1.4.3- بنية النموذج المعلم

قبل تصميم النموذج الطالب، من المهم فهم بنية النموذج المعلم. يعتمد نموذج **Depth Anything V2** على معمارية **محولات الرؤية (Vision Transformer - ViT)**، وتحديدًا نسخة **DINOv2**. يتميز هذا النموذج بقدرته على استخلاص سمات غنية وسياقية من الصورة بأكملها، مما يمنحه دقة فائقة في تقدير العمق. يتكون من مرمز (Encoder)

ضخم قائم على محولات الرؤية ومجموعة من الطبقات اللاحقة (Decoder) التي تقوم بتحويل هذه السمات إلى خريطة عمق. حجمه الكبير يجعله غير عملي للتطبيقات المحمولة. ولم يتم استخدام النموذج الكبير من Depth Anything لأن ذلك يصعب عملية التعلم على الطالب



الشكل 7- بنية النموذج المعلم

### 2.4.3- بنية النموذج الطالب

تم تصميم النموذج الطالب لتحقيق توازن بين الدقة وسرعة الاستدلال على الأجهزة المحمولة. مع مراعاة أنه عند تقطير المعرفة، من الأفضل تصميم بنية الطالب بحيث تحاكي بنية المعلم. تتكون بنية الطالب من جزأين رئيسيين:

#### 1. المرّمز:

تم اختيار معمارية **MobileViT-XS** كمرّمز للنموذج الطالب. يعتبر MobileViT نموذجاً هجيناً يجمع بين كفاءة الشبكات الالتفافية (CNNs) في استخلاص السمات المحلية وقدرة محولات الرؤية (Vision Transformer) على فهم السياق العام للصورة. هذه البنية تجعله خفيفاً وسريعاً ومثالياً للعمل على الهواتف المحمولة. تم استخدام النسخة المدربة مسبقاً على مجموعة بيانات ImageNet لاستخلاص السمات الأساسية.

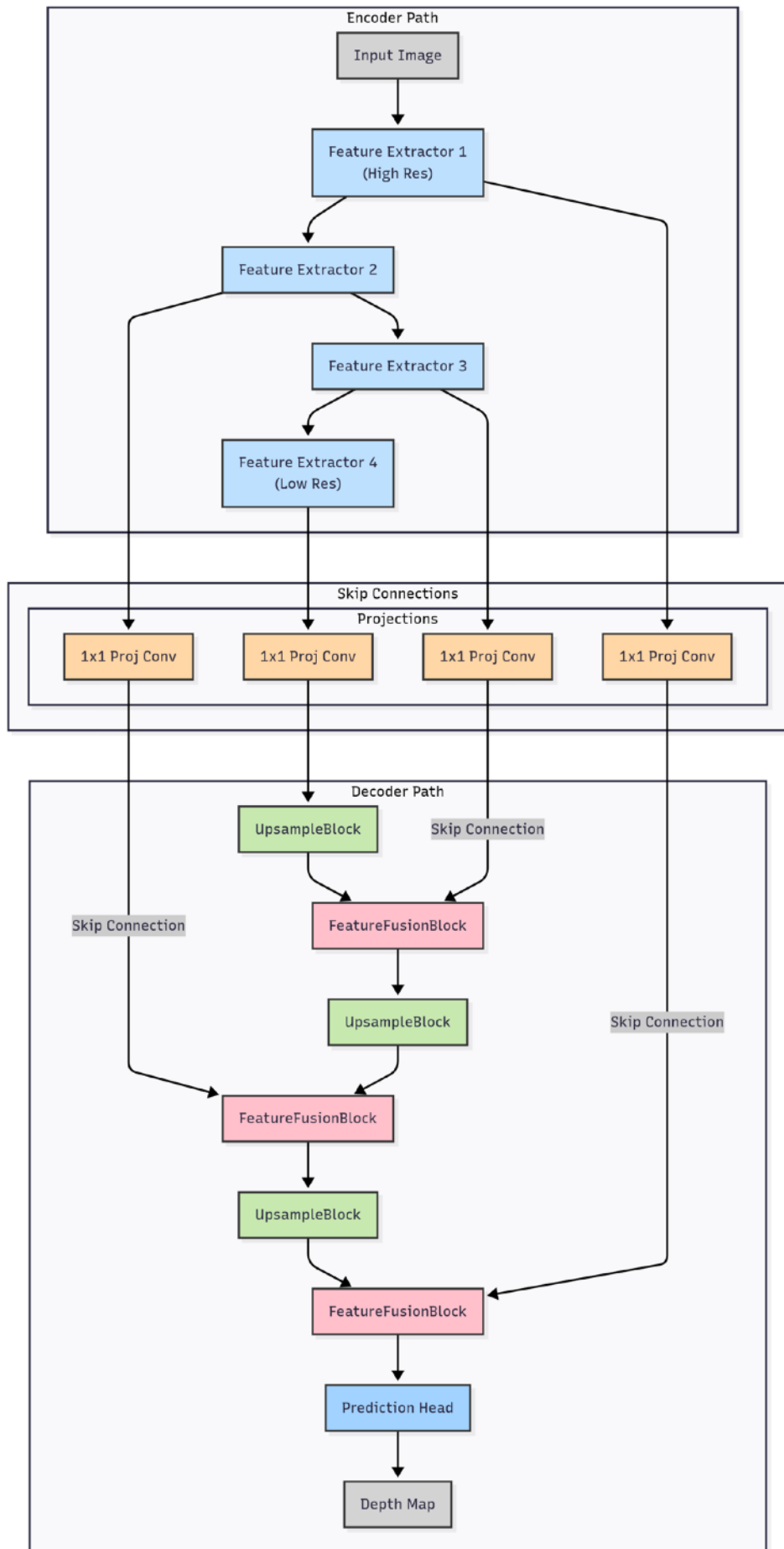
#### 2. مفك الترميز:

تم تصميم مفك ترميز مخصص وخفيف الوزن أطلقنا عليه اسم **MiniDPT**، وهو مستوحى من بنية DPT ولكن مع تبسيطات كبيرة لتقليل التعقيد الحسابي. يتلقى مفك الترميز خرائط السمات من مستويات متعددة من المرّمز، ويقوم بدمجها تدريجياً لإنتاج خريطة العمق النهائية. وتتم هذه العملية عبر ثلاث مراحل:

- **مرحلة الإسقاط (Projection Stage):** في البداية، يتم استخدام طبقات التفاف  $1 \times 1$  لتوحيد عدد القنوات في خرائط السمات القادمة من المرّمز، مما يجهزها لعملية الدمج.
- **مرحلة الرفع والدمج (Upsampling & Fusion):** هذه هي المرحلة الأساسية في مفك الترميز. تتم بشكل تكراري، حيث يتم في كل خطوة:
  - رفع أبعاد خريطة السمات من المرحلة السابقة (Upsample).
  - دمجها مع خريطة السمات المناظرة القادمة من المرّمز (skip-connection) باستخدام كتلة دمج مخصصة (Feature Fusion). هذا الدمج التدريجي يسمح للنموذج بالجمع بين المعلومات الدلالية عالية المستوى (من الطبقات العميقة) والتفاصيل المكانية الدقيقة (من الطبقات السطحية).
- **رأس التنبؤ (Prediction Head):** في النهاية، تقوم مجموعة من طبقات الالتفاف وطبقة رفع نهائية بمعالجة الخريطة المدمجة لإنتاج خريطة العمق النهائية بقناة واحدة، مع استخدام دالة (Sigmoid) لضمان أن تكون قيم الخرج بين 0 و 1.

هذا النموذج مكوّن من 3373265 متحول (3.4 مليون). بينما النموذج المعلم يتكون من 24.8 مليون، مما يعني أننا خفضنا النموذج بنسبة 1/7 تقريباً!





الشكل 8- بنية الطالب

## الفصل الرابع: التنفيذ والتطبيق

### 1.4- البيئات والأدوات المستخدمة

1. إطار العمل **Pytorch**: تم اختياره لمرونته وسهولة استخدامه في بناء وتدريب نماذج التعلم العميق، بالإضافة إلى الدعم الواسع من المجتمع العلمي والمصادر المتاحة.
2. **حزمة التطوير Flutter**: تستخدم لبناء تطبيق الهواتف المحمولة عبر منصتي **Android** و **iOS** من قاعدة كود واحدة، مما يسهل عملية التطوير والنشر. تم اختيارها بسبب وجود مكاتب تدعم تشغيل نماذج الذكاء الصناعي بصيغة **onnx**.
3. **محرك النصوص VS Code**: هو محرر نصوص متقدم يدعم الكثير من اللغات والميزات بسبب قدرته على تحميل لوائح متنوعة في كافة المجالات. وقد تم استخدامه في كتابة كل الرماز البرمجي.
4. **أداء التحكم في النسخ GIT**: هو نظام تحكم في النسخ لتعقب التغيرات التي تطرأ على البرنامج في مرحلة التطوير بحيث تسهل من عملية كشف الأخطاء والتراجع عنها، حيث تمكنا هذه الأداة من الاحتفاظ بنسخ من البرنامج أثناء عملية التطوير مع إمكانية العودة إلى أي منها في حال الحاجة إلى ذلك.
5. **البيئة ONNX (Open Neural Network Exchange)**: هو تنسيق ملفات يُستخدم لتشغيل نماذج الذكاء الاصطناعي المدربة. **Flutter** يحتوي على مكتبات تدعم تشغيل النماذج بهذا التنسيق، وهو أمر حاسم لنشر النموذج على جهاز محمول.
6. **النموذج Depth Anything V2 Large**: هو النموذج "المعلم" المحدد والمدرب مسبقاً الذي يُستخدم لتقديم الخبرة التي يتم "تقطيرها" إلى النموذج الطالب.
7. **بنية MobileNetV2**: هي معمارية شبكات عصبونية التلافيفية (CNN) عالية الكفاءة، مصممة خصيصاً للأجهزة المحمولة ومحدودة الموارد. تم النظر في استخدامها كمرمز (Encoder) للنموذج الطالب بفضل بنيتها خفيفة الوزن وقدرتها على تحقيق توازن جيد بين الدقة وسرعة الاستدلال، وذلك كإحدى المحاولات التي سبقت اعتماد بنية **MobileViT-XS** النهائية.
8. **بنية MobileViT-XS**: هي بنية الشبكة العصبونية المحددة التي تم اختيارها "كمرمز" للنموذج الطالب خفيف الوزن. وهي بنية هجينة تجمع بين كفاءة الشبكات الالتفافية وقدرة محولات الرؤية، مما يجعلها مثالية للأجهزة المحمولة.
9. **خدمة Google Colab**: هي بيئة تطوير سحابية قائمة على دقاتر **Jupyter**، تم استخدامها في هذا المشروع لتدريب نموذج التعلم العميق. أتاحت المنصة الوصول إلى موارد حاسوبية عالية الأداء، وتحديدًا وحدات معالجة الرسومات (GPU)، مما أدى إلى تسريع عملية تدريب النماذج وتجربة مختلف الأبنية والتكوينات بكفاءة ودون الحاجة إلى عتاد محلي باهظ الثمن.

### 2.4- بناء النموذج وتدريبه

#### 1.2.4- تهيئة البيئة

تم إنشاء بيئة افتراضية (Virtual Environment) وهي بيئة تحوي مفسر لغة **Python** والمكتبات والملفات المتعلقة بشكل مستقل ومعزول عن باقي البيئات الافتراضية في المشاريع الأخرى. أي بدل أن يكون هناك مفسر لغة واحد بنسخة ما ومكتبات بنسخ محددة واحدة لكل المشاريع، يصبح لكل مشروع بيئة افتراضية خاصة به تحوي مفسر لغة **Python** ومكتبات بنسخ تناسب المشروع بشكل مستقل عن باقي المشاريع، وهذا يجنبنا التعارض بينها. وبعد تفعيل البيئة تم تنزيل وتثبيت المكتبات المعرفة ضمن الملف **requirements.txt**، وهي ضرورية لتنفيذ الرماز البرمجي لبناء النموذج وتدريبه

#### 2.2.4- هيكل الملفات

لدينا البنية البرمجية التالية:

- **مجلد Config**: وهو مجلد يحوي جميع الثوابت والمعلومات العامة التي تفيد في جميع الملفات، مثل مسارات النماذج وأبعاد الصور المناسبة، والتحويلات المطبقة على الصور قبل التدريب والاختبار، وأوزان الترجيح المختارة لتابع الخسارة.
- **مجلد notebooks**: يحوي على دقاتر **Jupyter** تم تنفيذها على **Google Colab** بهدف تدريب واختبار بنية نموذج أو تابع خسارة معينة. بسبب عدم توافر موارد محلية.

• **مجلد Models:** يحوي هذا المجلد الرمز البرمجي لبنية نماذج الطالب والمعلم. من الجدير بالذكر أنه تم استخدام النمط التصميمي Factory بهدف استخدام بنى مختلفة للطالب أو ربما المعلم، مما يضمن قاعدة رماز نظيفة لا نحتاج سوا لإضافة سطرين ضمن ملف factory.py وتغيير سطر ضمن config.py وتبقى الملفات التي تستخدم النماذج بدون تغيير مهما غيرنا النماذج المستخدمة.

بالنسبة للمعلم فقد تم بناء صف يغلف النموذج المدرب بهدف تسوية خرجة واستخراج بعض الميزات الداخلية. حيث نقوم بتمرير الصورة للنموذج المدرب مسبقاً، ونستخلص بعض خرائط السمات من طبقات معينة لتكون أهدافاً يتعلمها الطالب، ثم نعيد تشكيل هذه السمات لصيغة يفهمها الطالب (حيث هناك فجوة تقنية بين طريقة فهم المحولات وطريقة فهم الشبكات الالتفافية)، وفي النهاية نقوم بتسوية خريطة العمق لتكون قيم بكسلاتها بين ال0 و ال1. أما الطالب فقد تم اختبار عدة بنى للطالب نذكر آخرها. وهو موزع على أربعة صفوف كالتالي:

1. الصف UpsampleBlock: يعرف كتلة الرفع المستخدمة في مفك الترميز. تقوم بمضاعفة الدقة المكانية (Spatial Resolution) للدخل وتستخدم الالتفافات العميقة القابلة للفصل (Depthwise Separable Convolutions) وهي تحقق كفاءة هائلة، فهي تقلل بشكل كبير من عدد العمليات الحسابية وحجم النموذج مقارنة بالالتفافات التقليدية. لذلك نستخدمها في حالتنا.

2. صف FeatureFusionBlock: يعرف كتلة الدمج المستخدمة في مفك الترميز. تقوم بدمج الميزات عالية المستوى من مرحلة مفك الترميز مع ميزات متدنية المستوى من مرحلة المرمز. هي التطبيق لوصلات التخطي في حالتنا.

3. صف MiniDPT: يمثل مفك الترميز. وهو يحاكي مفك الترميز المستخدم في النموذج المعلم ولكنه أخف لتحقيق هدفه. يقوم بإسقاط خرائط الميزات من المرمز لتناسب أبعادها مع مفك الترميز، ويعرف -بشكل تكراري- عدة طبقات من الكتلتين المعرفتين سابقاً. وفي النهاية نستخدم رأس التنبؤ مع تابع تفعيل Sigmoid لإصدار خريطة العمق النهائية بقيم بين 0 و 1.

4. صف StudentDepthModel: يستخدم المرمز المدرب مسبقاً (التصميم النهائي يستخدم MobileViT-XS) ويستخلص منه خرائط ميزات داخلية ويقوم بتمريرها لمفك الترميز، ومن ثم يصدر الخرج النهائي.

• **مجلد datasets:** يحتوي على صف لتحميل مجموعة البيانات للتدريب والاختبار، ويطبق عليها التحويلات المرجوة (Transforms).

• **مجلد data:** يحوي على نص برمجي يقوم بتسمية الصور بتسمية مرقمة، ومجلد images يحوي كل الصور التي تم تدريب النموذج عليها، وهي صور جمعتها بمساعدة بعض الأصدقاء.

• **مجلد criterion:** نعرف هنا توابع الخسارة المستخدمة. وبسبب طبيعة التدريب، حيث نحتاج لاختبار عدة توابع خسارة، تم استخدام نمطي التصميم Factory و Strategy ليسمح التصميم بتغيير تابع الخسارة بمرونة عن طريق تغيير سطر واحد في config.py وإضافة الاستراتيجية الجديدة. التابع الأساسي المستخدم معرف في الصف CombinedDistillationLoss وهو يطبق ما تم تعريف نظرياً في الفصل الثالث قسم منهجية التدريب.

• **مجلد utils:** يحتوي على أدوات مفيدة خلال عملية التدريب والاستخدام. حيث يحوي أيضاً على النمط التصميمي Factory ليسمح لنا باختيار معاملات تحسين (Optimizers) مختلفة. ونعرف هنا التحويلات للمطابقة على الصورة قبل إدخالها للنموذج.

يوجد أيضاً ملف metrics الذي يقوم بحساب معايير تقييم خريطة العمق حسب المعايير المذكورة في الدراسة النظرية. نستخدم هذا التابع في أثناء تقييم النموذج في النص البرمجي الخاص بالتقييم.

• **مجلد scripts:** يحتوي هذا المجلد على نصوص برمجية تقوم بعملية ما من أولها لآخرها (Complete pipeline) وهي:

1. نص التدريب يقوم بتحميل النماذج ومجموعة البيانات والقيام بالتدريب وحفظ أفضل نموذج تم الوصول إليه حسب قيمة تابع الخسارة على مجموعة خاصة بالتحقق (Validation).

2. نص برمجي خاص بالتقييم، يظهر أداء النموذج وفقاً للمعايير المعرفة في الدراسة النظرية.

3. نص برمجي خاص بالاستدلال () يقوم باستخدام النموذج المحدد لإصدار خرائط العمق لصورة او عدة صور محددة.

4. نص لتحويل النموذج من صيغة pth لصيغة onnx، يفيد عند مرحلة النشر/الإطلاق (Deployment).

• **مجلد tests:** هذا المجلد يحوي الاختبارات. حيث تم تعريف اختبار الوحدات (unit testing) لكل وحدة ضمن هذا البرنامج. حيث تم اختبار كل صف وكل تابع في حالات حدية مختلفة. ومن ثم تم اختبار التكامل (Integration Testing) بين هذه الوحدات، عن طريق اختبار خط إنتاج كامل (عملية تدريب مصغرة مثلاً).س

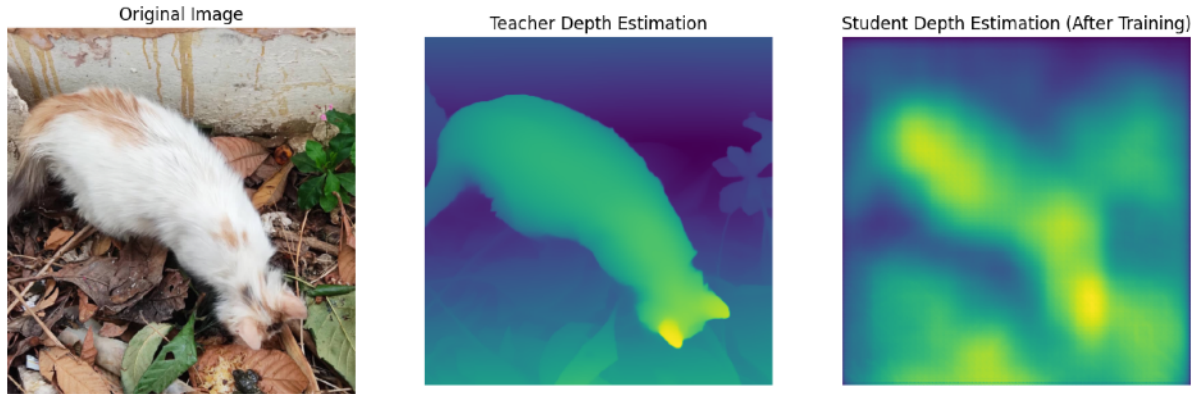
## 3.2.4- محاولات التدريب

تم إجراء عدة تجارب لتحسين أداء النموذج وتصميمه. سيتم التدريب في هذه المرحلة على مجموعة معطيات مصغرة لنجد النموذج المناسب ومن ثم وفيما يلي ملخص لأبرز هذه المحاولات:

### 1. المحاولة الأولى: استخدام المرمز MobileNetV3

**البنية:** المرمز mobilenet\_v3\_large مع مفك ترميز بسيط مكون من طبقات رفع وطبقات التفاضلية، وبأوزان ترجيح متماثلة لمكونات تابع الخسارة.

**الملاحظات:** كانت النتائج ضعيفة، حيث فقدت خريطة العمق الكثير من التفاصيل بسبب غياب السمات منخفضة المستوى أثناء عملية إعادة البناء. كما موضح في الصورة:



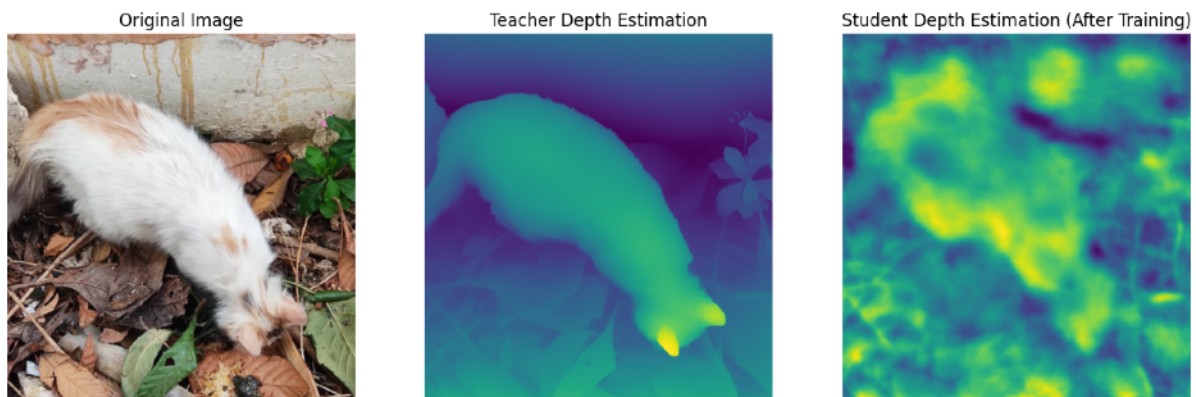
الشكل 9- نتائج محاولة التدريب الأولى

بعد هذه المحاولة تم تعديل أوزان الترجيح لمركبات الخسارة، زيادة وزن خسارة التدرج لتحفيز النموذج على التقاط الحواف وزيادة التركيز على خسارة SILog كي يهتم بالعمق بنفس قدر التدرج، أما بالنسبة للمركبتين المتبقيتين فتم تخفيضهما.

### 2. المحاولة الثانية: المرمز ذاته مع وصلات تخطي:

**البنية:** نفس بنية MobileNetV3 السابقة لكن مع إضافة وصلات تخطي من طبقات مختلفة في المرمز إلى الطبقات المقابلة في مفك الترميز. تم أيضاً تعديل أوزان الترجيح لدالة الخسارة حيث نحتاج لزيادة خسارة التدرج مع **الملاحظات:** لوحظ تحسن في جودة خرائط العمق، حيث ساهمت الوصلات في استعادة التفاصيل المكانية الدقيقة. ومع ذلك، كانت البنية لا تزال غير مثالية.

بعد هذه المحاولة تم ضبط الأوزان لتابع الخسارة عدة مرات وتغيير طبقات المرمز عدة مرات وكانت النتائج مشابهة. ولكن تشترك النتائج بالافتقار للسياق الشمولي (Global Context)، من هذه النقطة نتوجه لاستخدام MovileViT كمرمز.



الشكل 10- نتائج محاولة التدريب الثانية

### 3. المحاولة الثالثة: استخدام MobileViT:

**البنية:** المرمز MobileViT-XS ومفكك الترميز هو ذاته الموصوف في بنية الطالب (MiniDPT). وهي البنية الأخيرة والأفضل

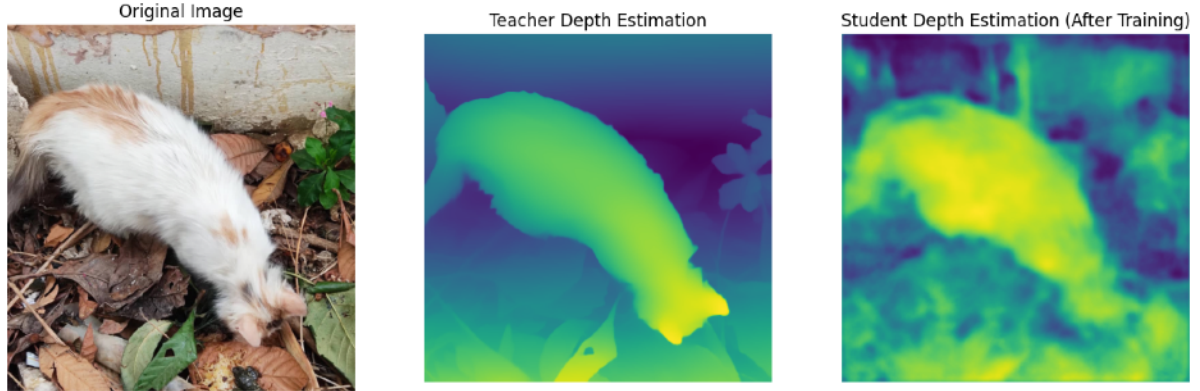
**تفاصيل التدريب:**

- **مجموعة البيانات:** صور غير موسومة مأخوذة بكاميرا جهاز محمول.
- **زيادة البيانات (Data Augmentation):** تم تطبيق عمليات قلب أفقي، دوران عشوائي، قص وتغيير حجم، وتعديل الألوان (Color Jitter) لزيادة تنوع البيانات ومنع فرط الملاءمة (Overfitting).
- **المحسن (Optimizer):** تم استخدام CosineAnnealingLR لضبط معدل التعلم بشكل ديناميكي خلال التدريب، مما يساعد في الوصول إلى نقطة تقارب أفضل.
- **تابع الخسارة:** المذكور سابقاً. مع أوزان كالتالي:

الوزن	$\lambda_{silog}$	$\lambda_{grad}$	$\lambda_{feat}$	$\lambda_{attn}$
القيمة	1.0	1.0	0.6	0.5

**الجدول 2- أوزان الترجية لمكونات تابع الخسارة**

- **عدد الحقب (Epochs):** تم التدريب لمدة 60 حقبة.
- الملاحظات:** نشهد تحسناً ملحوظاً في النتائج. فهم للسياق العام وقدرة على التقاط الحواف. ولكن النموذج ينخدع بالألوان ويعتقد أنها على عمق مختلف، كما في السائل على الحائط. يعود السبب لنقص البيانات في التدريب وغالباً ستحل المشكلة بزيادة البيانات.



**الشكل 11- نتائج محاولة التدريب الثالثة**

سنعتمد هذا النموذج ونقوم بعملية التدريب النهائية على قاعدة معطيات أكبر. — — — سيتم عرض تابع الخسارة لمرحلة التدريب الأخيرة عند إكمالها.

### 3.4- بناء تطبيق الموبايل

لإثبات الجدوى العملية للنموذج الطالب الذي تم تطويره، وتأكيداً على تحقيقه لمتطلبات الأداء على الأجهزة محدودة الموارد، تم بناء تطبيق عرض توضيحي (Proof of Concept) للهواتف المحمولة. كما ذكر مسبقاً، سيتم استخدام flutter لتطوير التطبيق.

#### 1.3.4- البنية الهيكلية

اتّبع التطبيق بنية تعتمد على فصل المسؤوليات لضمان سهولة الصيانة والتطوير المستقبلي. والبنية كالتالي:

- **إدارة الحالة (State Management):** تم استخدام حزمة Provider مع نمط ChangeNotifier لإدارة حالة التطبيق بشكل مركزي وفَعَال. هذا النهج يفصل منطق العمل (Business Logic) عن الواجهة الرسومية (UI)، حيث تقوم الواجهات بالاستماع للتغيرات في الحالة (AppState) وتحديث نفسها تلقائياً عند حدوث أي تغيير (مثل اختيار صورة جديدة أو انتهاء عملية المعالجة).
- **الخدمات (Services):** تم تغليف المنطق الأساسي في طبقة خدمات معزولة.
  - خدمة تحميل النموذج (Model Loader): خدمة مسؤولة عن تحميل نموذج ONNX من ملفات التطبيق. وتقوم هذه الخدمة بتفعيل مسرعات العتاد (Hardware Acceleration) المتاحة في نظام التشغيل لضمان أقصى أداء ممكن، حيث تستخدم NNAPI على أجهزة Android و CoreML على أجهزة iOS.
  - خدمة تقدير العمق (Depth Estimator): تحتوي هذه الخدمة على المنطق الكامل لعملية تقدير العمق، بدءاً من المعالجة المسبقة للصورة وحتى تطبيق خرائط الألوان على المنطق الكامل لعملية تقدير العمق، بدءاً من أدوات مفيدة (Utils): بعض الأدوات التي تحتاجها الخدمات
- **الواجهات الرسومية (User Interface):** تم بناء واجهة المستخدم من مجموعة من الوحدات (Widgets) المستقلة والقابلة لإعادة الاستخدام، واستخدام الوحدات في شاشات العرض (screens)، مما يجعل الرمان البرمجي أكثر تنظيماً ووضوحاً.

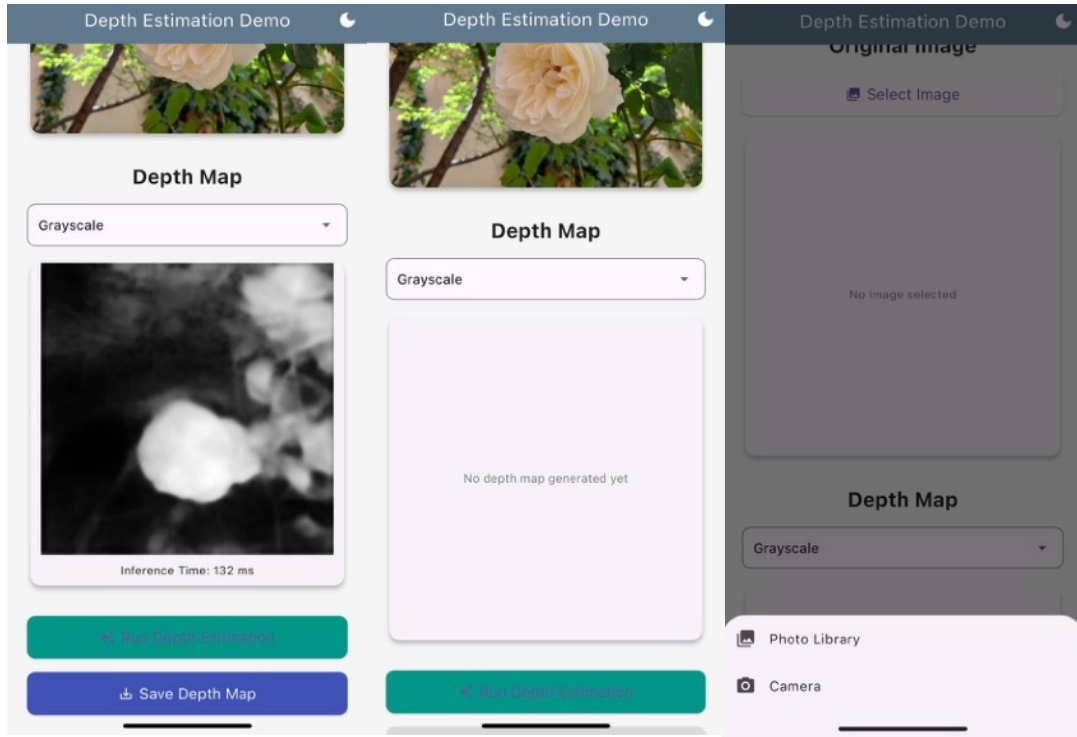
## 2.3.4- تحسينات الأداء

- لتحقيق متطلب الأداء الأساسي وهو الاستجابة في الزمن الحقيقي وتجنب تجميد واجهة المستخدم أثناء العمليات الحاسوبية المكثفة، تم تطبيق تحسينين رئيسيين:
1. استخدام **onnxruntime**: تم الاعتماد على هذه المكتبة المتخصصة لتشغيل نماذج الذكاء الاصطناعي على الأجهزة المحمولة، حيث توفر أفضل أداء ممكن من خلال الاستفادة من قدرات العتاد المحلي.
  2. **المعالجة في الخلفية (Background Isolates):** إن عمليتي المعالجة المسبقة للصورة (تغيير الحجم والتطبيع) والمعالجة اللاحقة لخريطة العمق (تطبيق خريطة الألوان) هما عمليتان مكثفتان حاسوبياً. لتجنب إعاقة خيط المعالجة الرئيسي (Main Thread) المسؤول عن سلاسة الواجهة الرسومية، تم نقل هاتين العمليتين ليتم تنفيذهما في خيوط معالجة منفصلة في الخلفية (Isolates) باستخدام دالة Compute التي يوفرها Flutter. يضمن هذا الأسلوب بقاء التطبيق سريع الاستجابة بشكل كامل حتى أثناء معالجة الصور.

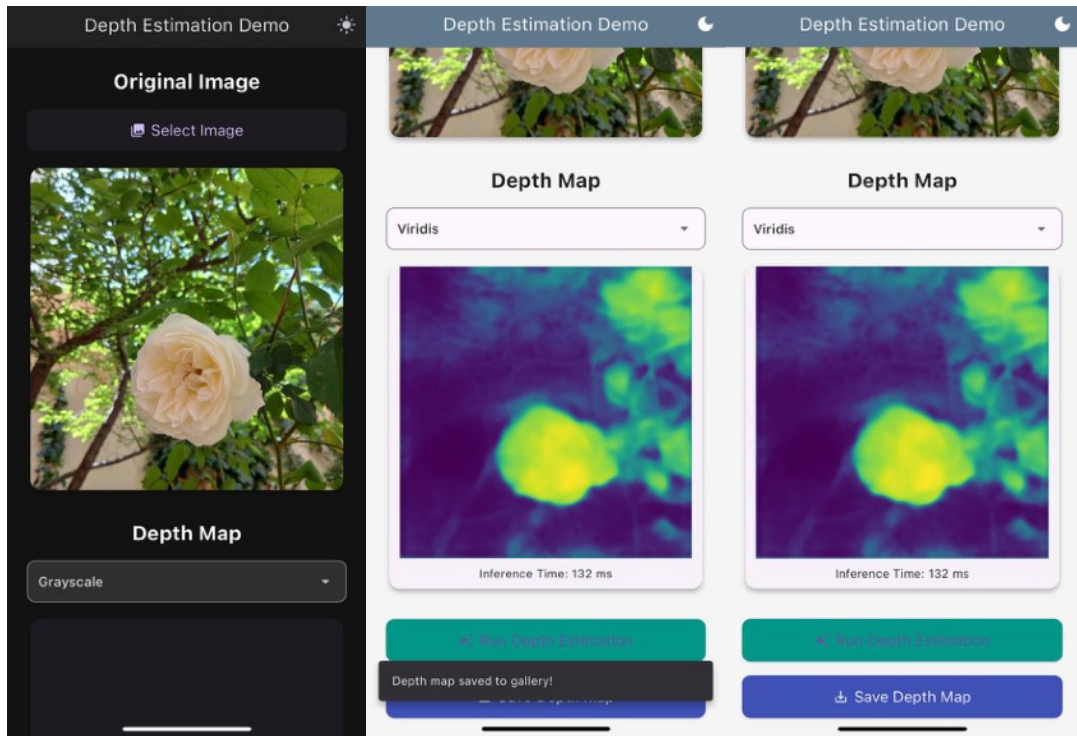
## 3.3.4- استعراض وظائف التطبيق

- \*ملاحظة: النتائج التي ستعرض أولية، ما زال النموذج قيد التدريب وسيتم عرض النتائج النهائية في التقرير النهائي.
- يوفر التطبيق واجهة بسيطة وبديهية تتيح للمستخدم تحقيق الوظائف التالية:
- **اختيار صورة:** يمكن للمستخدم التقاط صورة جديدة باستخدام الكاميرا أو اختيار صورة موجودة مسبقاً من معرض الصور.
  - **تقدير العمق:** يقوم زر مخصص ببدء عملية تقدير العمق، مع إظهار مؤشر "جاري المعالجة..." لتزويد المستخدم بتغذية راجعة مرئية.
  - **عرض النتائج:** يتم عرض الصورة الأصلية وخريطة العمق الناتجة على نفس الشاشة للمقارنة الفورية.
  - **قياس الأداء:** يتم عرض زمن الاستدلال (Inference Time) الذي استغرقه النموذج بالمللي ثانية أسفل خريطة العمق، مما يوفر دليلاً كمياً ومباشراً على كفاءة النموذج.
  - **تخصيص العرض:** يمكن للمستخدم تغيير خريطة الألوان المطبقة على الخرج (مثلاً، من تدرج رمادي إلى Viridis) للحصول على تصور أفضل للفروقات في العمق.
  - **نمط ليلي:** يوجد أيضاً دعم للنمط الليلي. نوضح تلك الوظائف بالأشكال التالية:





الشكل 12- اختيار صورة وتقدير العمق وعرض النتائج



الشكل 13- تخصيص العرض وحفظ الصورة والنمط الليلي

نلاحظ أن زمن الاستدلال لا يتجاوز ١٥٠ ميلي ثانية! وهذا أحد أهدافنا. بذلك، ينجح التطبيق في إثبات فعالية النموذج المقطّر وقابليته للعمل بكفاءة في بيئة تشغيل حقيقية على الهواتف المحمولة، محققاً الأهداف والمتطلبات التي تم تحديدها في بداية المشروع.

## الفصل الخامس: النتائج والمناقشة

### 1.5- بيئة الاختبار

تم تقييم النموذج على الأجهزة التالية:

- هاتف محمول (Iphone 13): بمعالج A15.
- حاسب محمول (Macbook Pro 2018): بمعالج Intel Core i7.
- بيئة Google Colab (المجانية): بكرت رسومات Tesla 4 GPU.

### 2.5- التقييم الكمي

تم تقييم أداء النموذج الطالب النهائي على مجموعة اختبار من KITTI، وتمت مقارنته بالنموذج المعلم. يوضح الجدول التالي النتائج المحققة وفقاً للمعايير المذكورة في الدراسة المرجعية. (ما زال النموذج بحاجة تدريب، هذه القم ليست القيم النهائية وسيتم تعديلها بعد انجاز التدريب)

النموذج	Abs Rel (↓)	Sq Rel (↓)	RMSE (↓)	RMSE Log(↓)	$\delta_1$ (↑)	$\delta_2$ (↑)	$\delta_3$ (↑)
المعلم	0.2828	0.0287	0.0593	0.3714	0.7309	0.8826	0.9412
الطالب	0.6435	0.1583	0.3062	1.0896	0.0615	0.1446	0.2686

الجدول 3- التقييم الكمي للنموذج الطالب

### 3.5- التقييم النوعي

تم اختبار النموذج على صور متنوعة. تُظهر الصور التالية مقارنة بين الصورة الأصلية، خريطة العمق من المعلم، وخريطة العمق من الطالب. النتائج الحالية ليست النهائية. يمثل الشكل التالي أداء النموذج الطالب (على اليسار) والنموذج المعلم (في الوسط) والصورة الأصلية على اليمين



الشكل 14- مقارنة بصرية



## 4.5- قياسات الأداء

تم قياس الأداء على الأجهزة المذكورة في بيئة الاختبار وكانت النتائج كالتالي:  
(لم نتمكن من تشغيل المعلم على الهاتف المحمول! لا تدعم بيئة المحمول العمليات المستخدمة في بنية المعلم)

النموذج	الزمن (ميلي ثانية)
المعلم على الحاسب المحمول	1124
الطالب على الحاسب المحمول	576.8
المعلم على Google Colab	60
الطالب على Google Colab	13.2
الطالب على الهاتف المحمول	137.1

الجدول 4- قياس الأداء للنموذج

حققنا زمن وسطي ١٣٧ ميلي ثانية على الهاتف المحمول مما يعني اننا حققنا الهدف المنشود.  
هناك انخفاض في الجودة وهذا معقول بسبب نقص التدريب حتى هذه اللحظة

## 5.5- المناقشة

لقد نجح هذا المشروع في تحقيق هدفه الأساسي المتمثل في إثبات فعالية تقنية تقطير المعرفة كاستراتيجية لتطوير نماذج تقدير عمق عالية الكفاءة وقابلة للتشغيل على الأجهزة المحمولة. حيث أثبتت النتائج أن النموذج الطالب الذي تم تطويره قادر على العمل في الزمن الحقيقي ضمن بيئة محدودة الموارد، محققاً بذلك التوازن الصعب بين الدقة الحاسوبية ومتطلبات الأداء. إن تحقيق زمن استدلال وسطي يبلغ حوالي 137 ميلي ثانية على هاتف محمول يفى بالمتطلب غير الوظيفي الذي تم تحديده بالألا يتجاوز 200 ميلي ثانية ، ويؤكد على نجاح النهج المتبع.

عند تحليل النتائج، نلاحظ وجود فجوة متوقعة في الدقة بين النموذج المعلم والنموذج الطالب، كما يتضح من التقييم الكمي على مجموعة بيانات KITTI. يمثل هذا الانخفاض في الدقة تضحية مقصودة ومقبولة في مقابل الحصول على مكاسب هائلة في الأداء. لقد تم تقليص حجم النموذج بنسبة تقارب 1/7 مقارنة بالمعلم ، والأهم من ذلك، تم تمكين القدرة على تشغيله على جهاز محمول، وهو أمر لم يكن ممكناً مع النموذج المعلم الأصلي. تُظهر المقارنات النوعية أن النموذج الطالب ينجح في النقاط الهيكل العام للمشاهد وتدرجات العمق الأساسية، على الرغم من فقدان بعض التفاصيل الدقيقة التي يحتفظ بها المعلم. هذه الجودة تعتبر كافية للعديد من التطبيقات العملية مثل الواقع المعزز ومساعدات التنقل الأساسية. يمكن إرجاع نجاح النموذج إلى عدة قرارات منهجية مدروسة. كان اختيار معمارية MobileViT-XS كمرمز قراراً حاسماً، حيث أثبتت قدرتها على فهم السياق الشمولي للصورة، متغلباً على قصور بنية MobileNetV3 الذي لوحظ في المحاولات الأولية. علاوة على ذلك، لعبت دالة الخسارة المركبة دوراً محورياً في عملية التقطير ؛ فمن خلال إجبار الطالب على محاكاة تدرجات وخرائط سمات المعلم، وليس فقط الخرج النهائي، تم نقل "خبرة" أعمق ساهمت في الحفاظ على جودة النتائج على الرغم من صغر حجم النموذج.

مع ذلك، لا يخلو المشروع من بعض المحددات التي يمكن معالجتها في المستقبل. أبرز هذه المحددات هو اعتماد النموذج على بيانات التدريب، حيث لوحظ أن النموذج قد ينخدع بألوان معينة في المشهد، معتقداً أنها على عمق مختلف. يُعزى هذا السلوك على الأرجح إلى محدودية وتنوع مجموعة البيانات المستخدمة في التدريب النهائي.

## 6.5- آفاق مستقبلية

بناءً على هذه المناقشة، يمكن تحديد عدة آفاق مستقبلية واعدة:

- **تحسين بيانات التدريب:** الخطوة التالية الأكثر أهمية هي إعادة تدريب النموذج على مجموعات بيانات عامة وأكثر تنوعاً (مثل NYU Depth V2) لتعزيز قدرته على التعميم وتقليل تأثره بالتحيزات الموجودة في البيانات الحالية.

- **تحسينات معمارية إضافية:** يمكن استكشاف معماريات هجينة أخرى أو تحسين بنية مفكك الترميز MiniDPT لتقليل فجوة الدقة مع المعلم دون زيادة كبيرة في التكلفة الحسابية.
- **تكميم النموذج (Model Quantization):** كخطوة أخيرة للتحسين، يمكن تطبيق تقنيات التكميم (مثل تحويل الأوزان إلى bit-16 أو bit integer-8) على نموذج ONNX النهائي. من شأن هذا الإجراء أن يقلل من حجم النموذج بشكل أكبر ويسرّع من زمن الاستدلال على الأجهزة المحمولة الداعمة.

## المراجع

— لاحقاً في التقرير النهائي —