

Vision Transformers

Docentes:

Esp. Abraham Rodriguez - FIUBA

Mg. Oksana Bokhonok - FIUBA

Programa de la materia

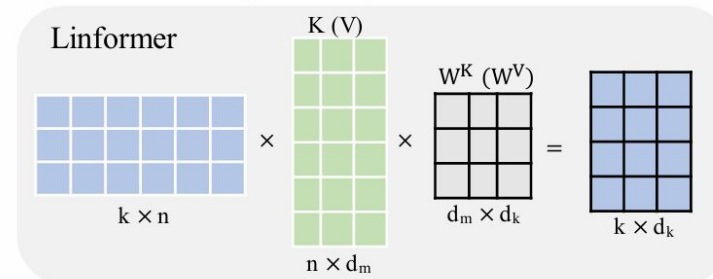
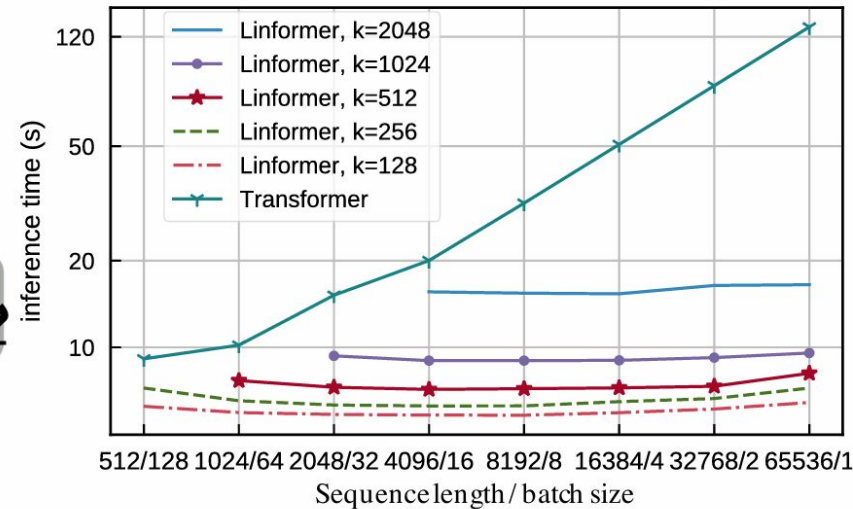
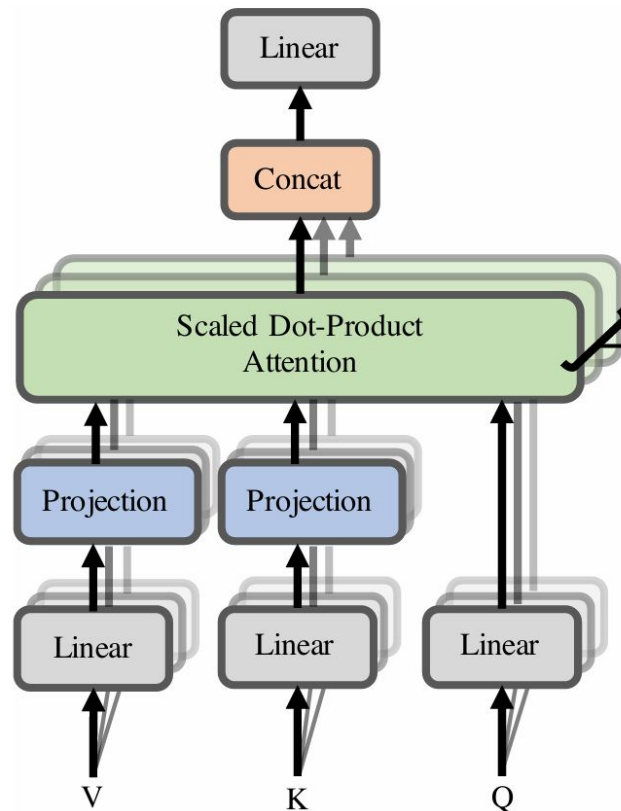
1. Arquitectura de Transformers e imágenes como secuencias.
2. Arquitecturas de ViT y el mecanismo de Attention.
3. Ecosistema actual, Huggingface y modelos pre entrenados.
4. GPT en NLP e ImageGPT.
5. Modelos multimodales: combinación de visión y lenguaje
6. Segmentación con SAM y herramientas de auto etiquetado multimodales.
7. OCR y detección con modelos multimodales.
8. Presentación de proyectos.

Repaso de algunas de las variantes de Attention

Linear Self-Attention

Simplificación de la autoatención para reducir el tiempo de cálculo

Eficiencia escalable para procesar imágenes de alta resolución

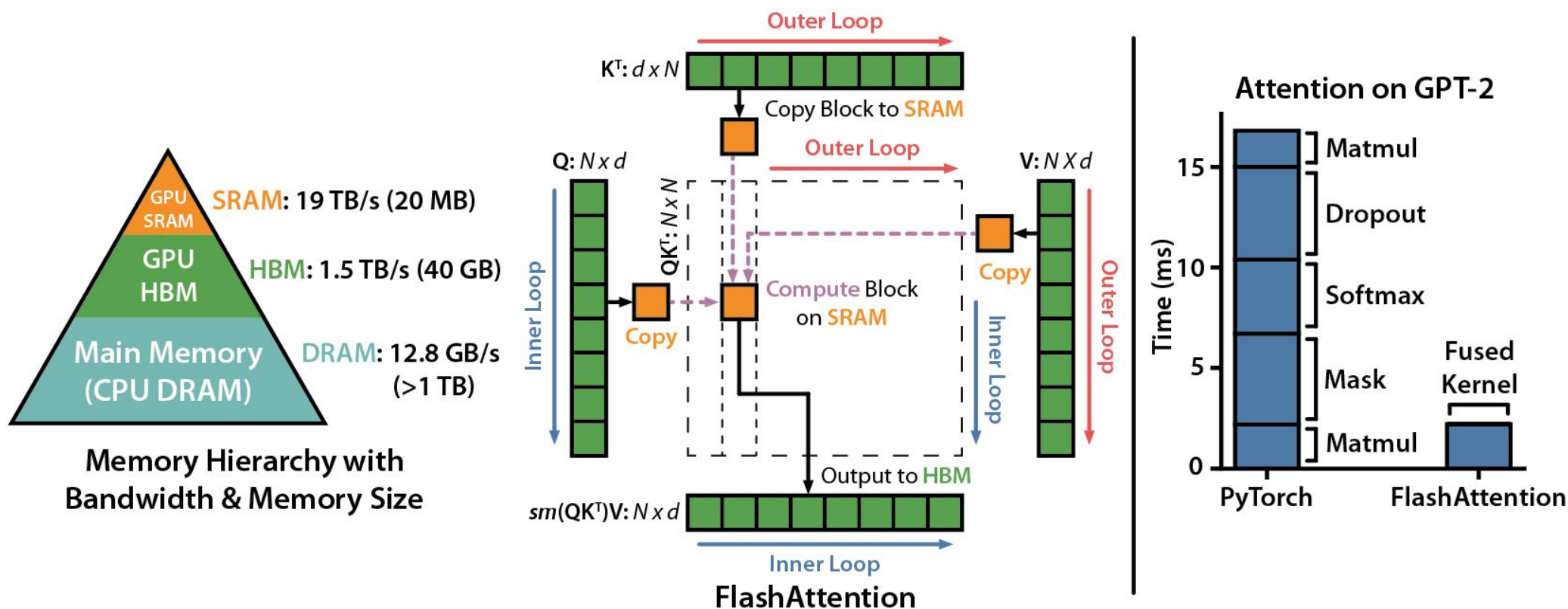


Repaso de algunas de las variantes de Attention

FlashAttention (estándar)

Optimizaciones en GPU para reducir el costo computacional de la autoatención estándar

Maneja eficientemente grandes secuencias con mejoras en velocidad y memoria

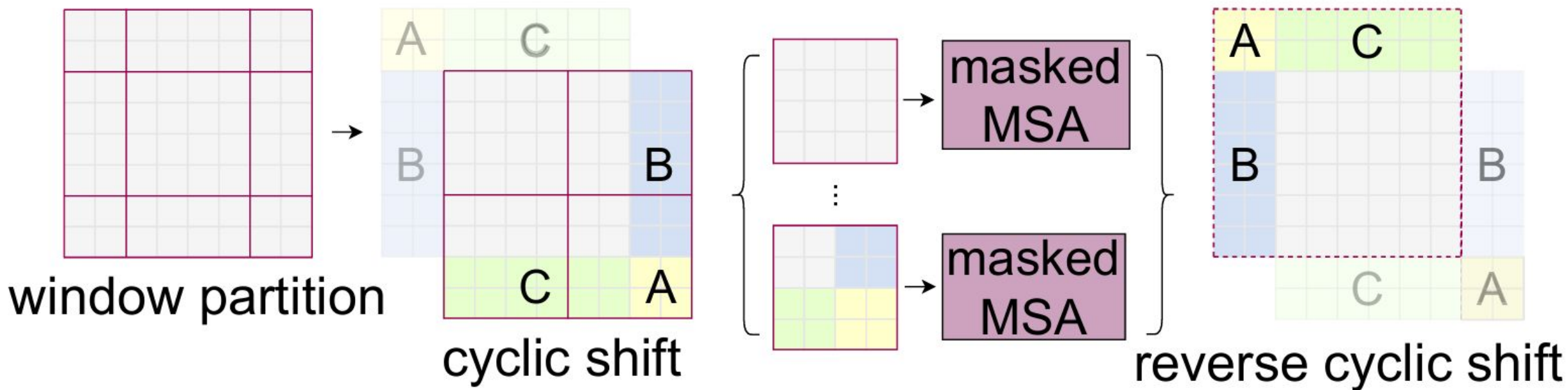


Repaso de algunas de las variantes de Attention

Window Multi-head Self Attention (Se usa en Swin Transformer)

Atención en ventanas locales para limitar el alcance y mejorar la eficiencia

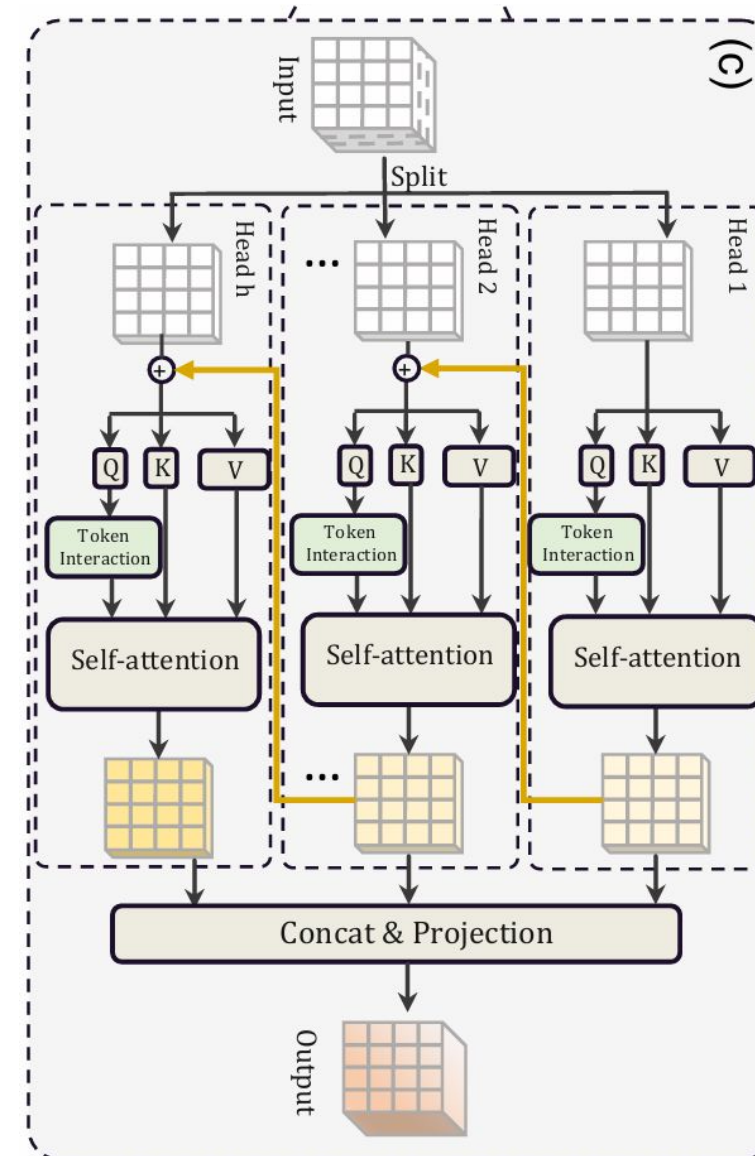
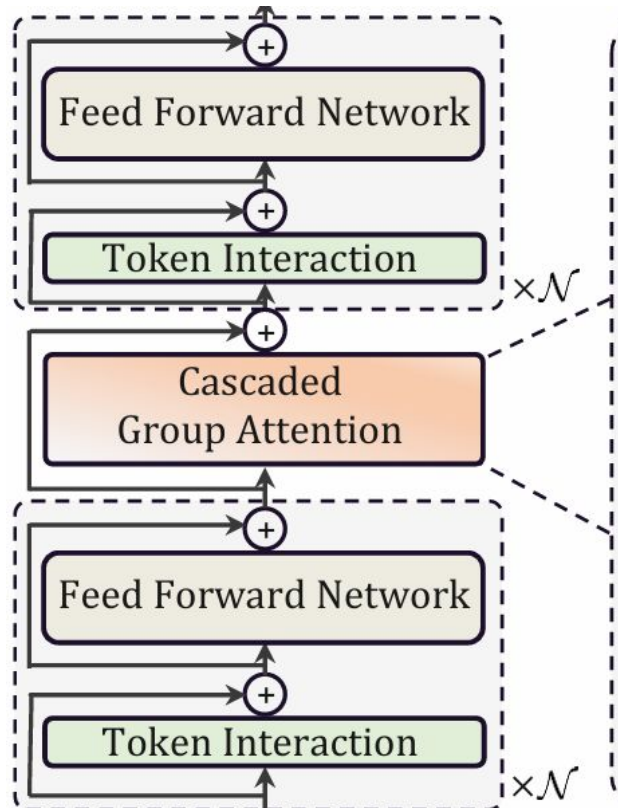
Permite capturar características locales en cada ventana de la imagen



Repaso de algunas de las variantes de Attention

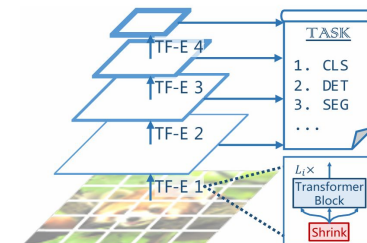
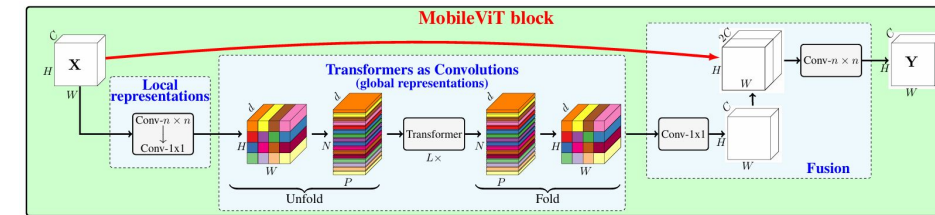
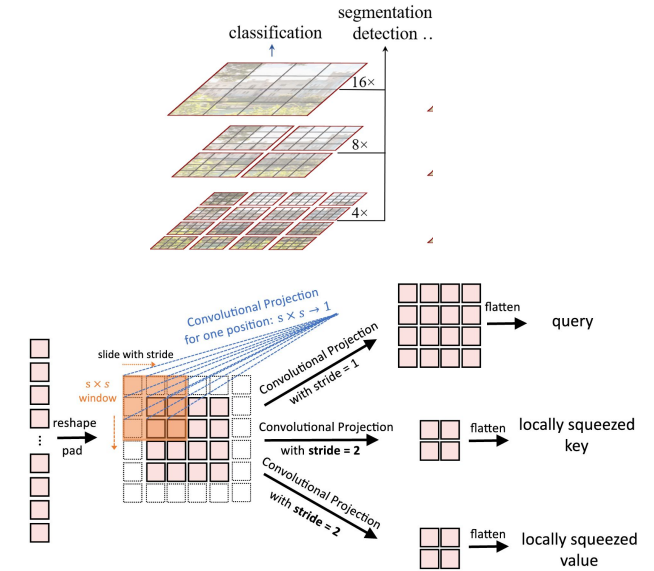
Cascaded Group Attention

Agrupa tokens en jerarquías para mejorar el enfoque local-global



Repaso de algunas de las arquitecturas ViT

- Swin Transformer ([Link-paper](#), [Link-huggingface](#))
 - Mecanismo de atención **jerárquica basado en ventanas deslizantes**.
 - Divide la imagen en ventanas no superpuestas para aplicar self-attention local.
 - Usa ventanas deslizantes para capturar información a mayor escala.
 - Más eficiente en memoria y mejor en la captura de detalles locales y globales que ViT clásico.
- Convolutional Vision Transformer (CvT) ([Link-paper](#), [Link-huggingface](#))
 - CvT promete incrementar el rendimiento y robustez de ViT mientras se conserva una alta eficiencia computacional. Introduce convolución. en dos partes de ViT:
 - Reemplaza la proyección lineal por proyección convolucional.
 - Utiliza una estructura jerárquica en múltiples etapas similar a CNNs.
- MobileViT ([Link-paper](#), [Link-huggingface](#))
 - combina CNNs y ViTs **para tareas de visión en dispositivos móviles** y de bajo consumo.
 - Sustituye el procesamiento local de las convoluciones con procesamiento global usando transformers.
- Pyramid ViT ([Link-paper](#), [Link-huggingface](#))
 - A diferencia de ViT, PVT genera salidas de alta resolución con menores costos computacionales y de memoria.
 - Combina ventajas de CNNs y Transformers, convirtiéndose en un backbone unificado para diversas tareas de visión.
 - Utiliza una pirámide progresiva y atención con reducción espacial para mejorar la resolución bajo recursos limitados.



ViTs. Ejemplos de uso

- Swin Transformer ([Link](#))

[Clasificación de imágenes](#)

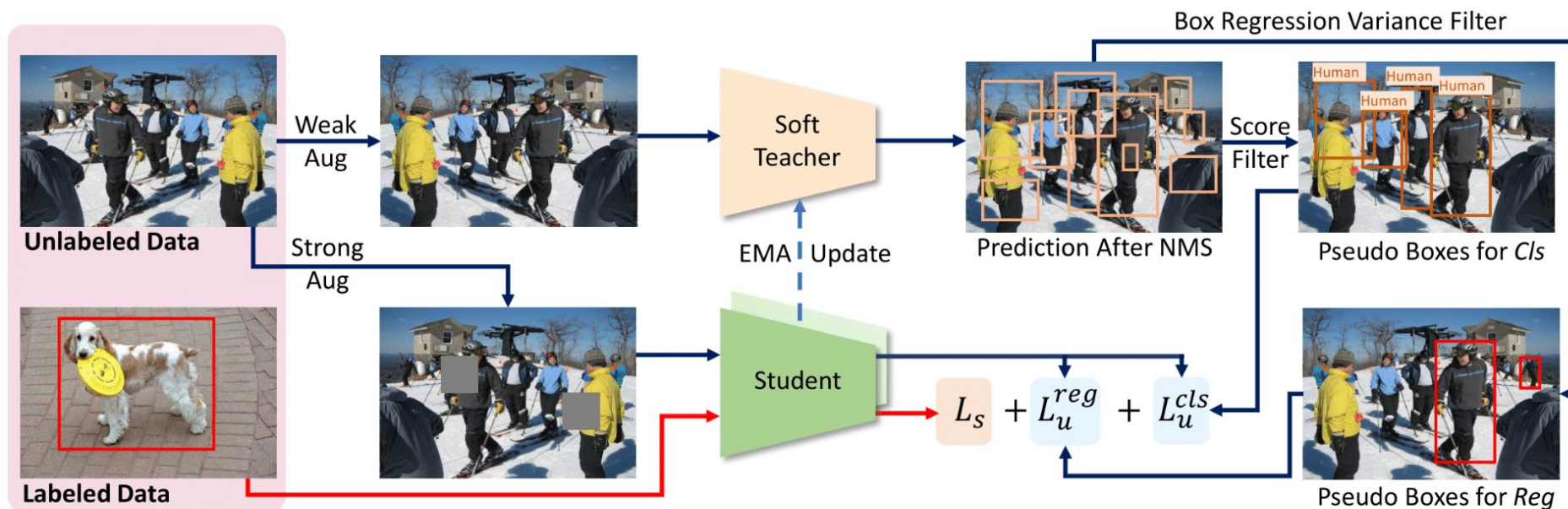
[Detección de objetos](#)

[Detección de objetos semisupervisado](#)

[Segmentación semantica](#)

[Reconocimiento de acciones en videos](#)

[Transformer-SSL: aprendizaje auto-supervisado contrastivo](#)



Ecosistema actual

Donde estan los ViT?

ViT es bastante nuevo (Oct 2020), respecto a los transformers, el **foco** está en LLMs y AI generativa donde virtualmente no tienen competencia (para Mayo 2020, ya existía **GPT-3**), pero en Visión Artificial, ViT tiene competencia, siendo las CNNs.

Sin embargo hoy en día podemos encontrar ViT en:

- Sistemas ADAS

- Robótica

- Sistemas Embebidos

OpenPilot

[OpenPilot](#) de comma.ai, es un proyecto que ha utilizado en su mayoría CNNs, hasta [recientemente](#), donde sustituyeron EfficientNet (CNN) por FastViT (CNN + ViT) de Apple.

[Website](#)

[Video](#)

OpenPilot Hardware

Una de las razones por las cuales hay algo de escepticismo respecto a ViT, **es el alto requerimiento de hardware del Transformer**, pero que hardware utiliza OpenPilot?

Cameras: Three 1080p cameras with 140 dB dynamic range, including dual-cam 360° vision and a narrow cam for distant objects.

Processor: **Qualcomm Snapdragon 845 (2017)** *Samsung galaxy S9, Pixel 3, y celulares de 5+ años!*

CAN FD Enabled: Supports CAN FD vehicles without extra hardware.

Storage: 128GB built-in.

Connectivity: LTE, Wi-Fi, and High-Precision GPS.

Night Vision: IR LEDs for interior night-vision monitoring.

Display: 2160x1080 OLED.

Ports: OBD-C (USB-C with CAN) and USB 3.1 Gen 2.



Apple Products

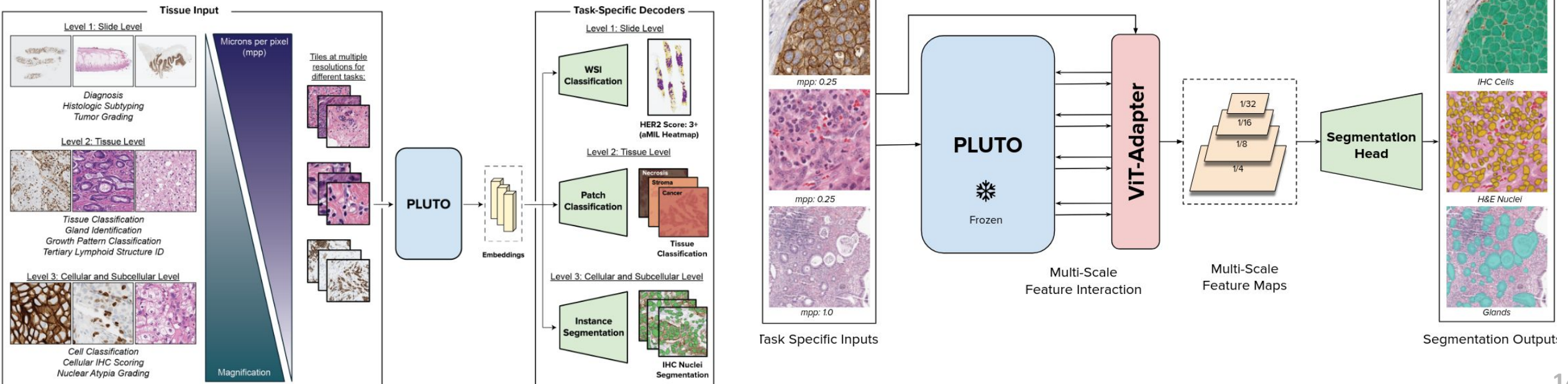
Apple esta a la vanguardia con respecto a arquitecturas ViT con MobileViT, MobileCLIP y FastViT.

[Deploying ViT to Apple Neural Engine](#)

PathAI (patología)

PathAI es una empresa de analisis de patologia utilizando IA, en Mayo 2024, publicaron el paper “[PLUTO: Pathology-Universal Transformer](#)”

A. PLUTO framework for multi-resolution pathology tasks

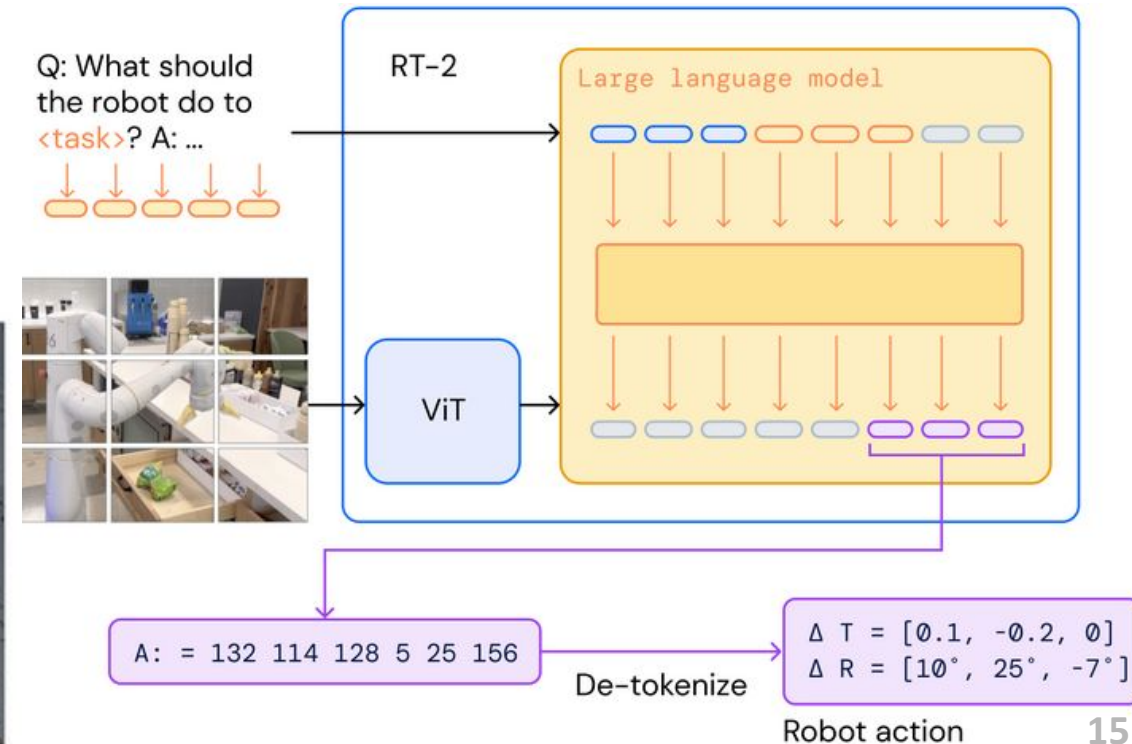


Robotic Transformer 2

RT-2 es un modelo Vision-Language (ViT y LLM) aplicado a robótica, el cual permite instruir a un robot a realizar tareas sin necesidad de conocimiento previo sobre las mismas, solamente con un prompt y una cámara.

Demostración.

Push the ketchup to the blue cube



Cloud Comparison Cheat Sheet

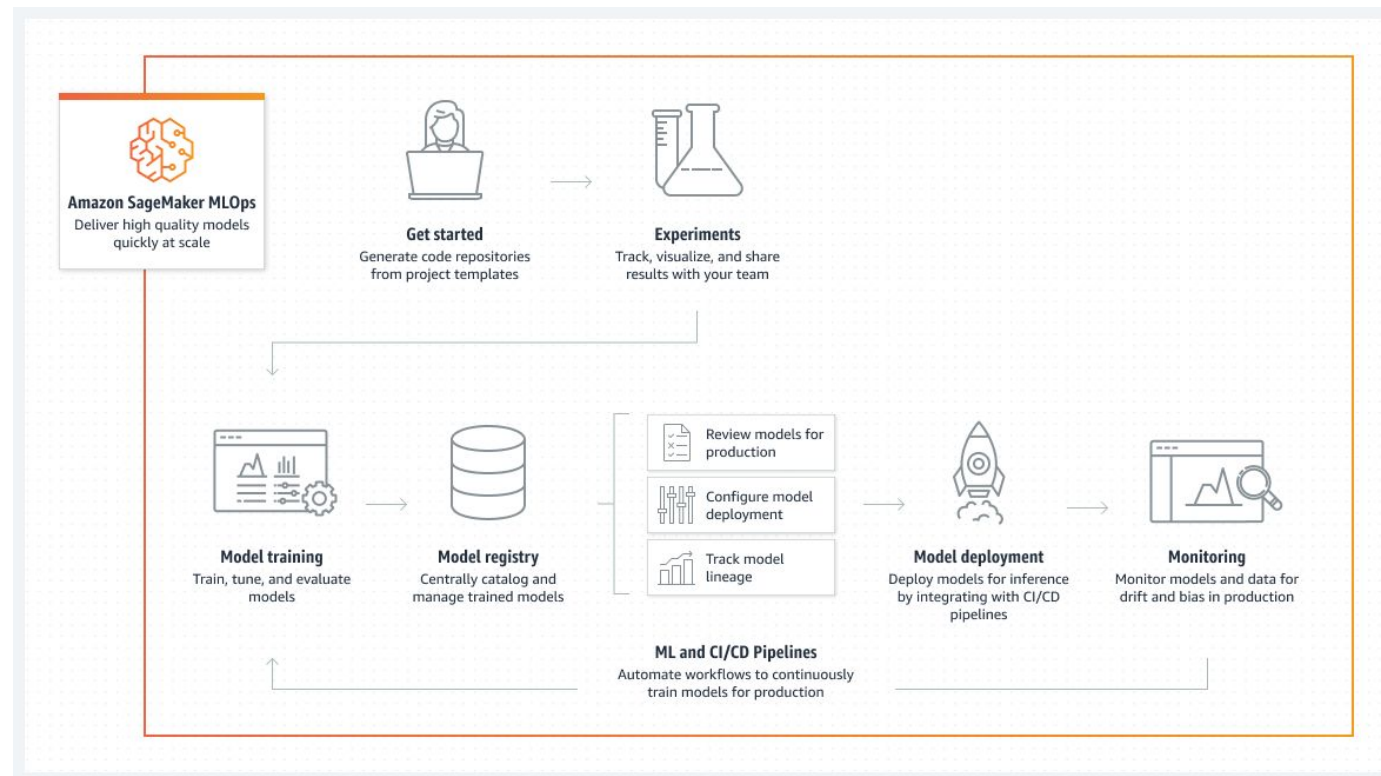
blog.bytebytego.com

aws	Azure	Google Cloud	ORACLE CLOUD
<ul style="list-style-type: none">Elastic Compute Cloud (EC2)Elastic Kubernetes Service (EKS)LambdaSimple Storage Service (S3)Elastic Block StoreElastic File SystemVirtual Private CloudRoute 53Elastic Load BalancingWeb Application FirewallRDSDynamoDBRedshiftElastic MapReduceKinesisSageMakerGlueEventBridgeSimple Queuing ServiceSimple Notification ServiceCloudWatchCloudFormationIAMKMS	<ul style="list-style-type: none">Virtual MachineAzure Kubernetes Service (AKS)Azure FunctionsBlob StorageManaged DiskFile StorageVirtual NetworkDNSLoad BalancerWeb Application FirewallSQL DatabaseCosmos DBSynapse AnalyticsHDInsightStreaming AnalyticsMachine LearningData FactoryEvent GridStorage QueuesService BusMonitorResource ManagerActive DirectoryKey Vault	<ul style="list-style-type: none">Compute EngineGoogle Kubernetes Engine (GKE)Cloud FunctionsCloud StoragePersistent DiskFile StoreVirtual Private CloudCloud DNSCloud Load BalancingCloud ArmorCloud SQLFirebase Realtime DatabaseBigQueryDataprocDataflowVertex AIData FusionEventarcPub/SubFirebase Cloud MessagingCloud MonitoringDeployment ManagerCloud IdentityCloud KMS	<ul style="list-style-type: none">Virtual MachineOracle Container EngineOCI FunctionsObject StoragePersistent VolumeFile StorageVirtual Cloud NetworkDNSLoad BalancerWeb Application FirewallATPNoSQL DatabaseAutonomous Data WarehouseBig DataStreamingData ScienceData IntegrationEventsStreamingNotificationsMonitoringResource ManagerIAMVault

SageMaker	Machine Learning	Vertex AI	Data Science
SageMaker Studio Lab Free Cloud Computing Services - AWS Free Tier Machine Learning Service - Free Amazon SageMaker - AWS	Crear tu cuenta gratuita de Azure hoy mismo Microsoft Azure	Free cloud features and trial offer Google Cloud Free Program	Oracle Cloud Free Tier Oracle
Servicios populares gratuitos durante 12 meses	Servicios populares gratuitos durante 12 meses	90 dias	Trial 4700 Horas
Servicios adicionales que siempre son gratuitos	55 servicios adicionales que siempre son gratuitos	Servicios adicionales que siempre son gratuitos	
SageMaker-2 meses	Comenzar con un crédito de Azure de 200 USD	Comenzar con un crédito de Google de 300 USD	
AI Courses and Training - Learn Artificial Intelligence - AWS	Microsoft Certified: Azure AI Fundamentals - Certifications Microsoft Learn	Machine Learning & AI Courses Google Cloud Training	Discover AI: Training for Beginners and Beyond - Oracle MyLearn

Amazon Sagemaker

- Servicio de aprendizaje automático de AWS, lanzado en 2017.
- Proporciona herramientas para construir, entrenar y desplegar modelos de aprendizaje automático.
- Incluye [SageMaker Studio](#) (IDE) y [MLOps](#) entre otros [servicios](#)



Amazon Sagemaker

[SageMaker Studio Lab](#)



My project

CPU and GPU runtime limits have changed.




You can use CPU for up to 4 hours at a time with a limit of 8 hours in a 24-hour period.
You can use GPU for up to 4 hours at a time with a limit of 4 hours in a 24-hour period.




Runtime status

Idle

Runtime remaining 

Session: —

Today: 8 h 0 m

Compute type 

☒ CPU ☐ GPU

▶ Start runtime

Open
project

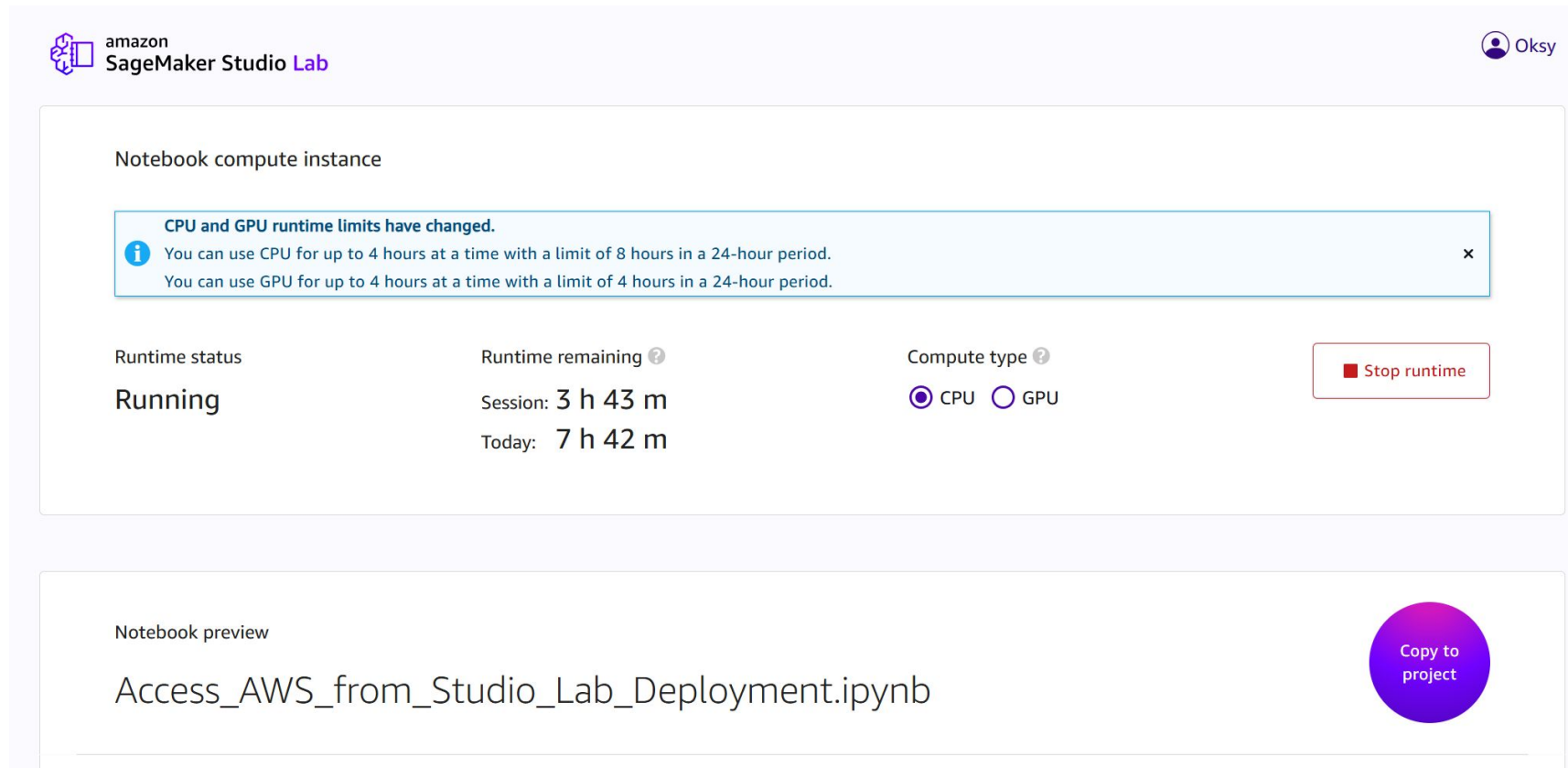
Amazon Sagemaker

[Hugging Face on Amazon SageMaker](#)

[Hugging Face Pretrained Model to Amazon SageMaker](#)

[Amazon SageMaker](#)

[Hugging Face — sagemaker 2.232.2 documentation](#)



The screenshot displays the Amazon SageMaker Studio Lab interface. At the top left is the "amazon SageMaker Studio Lab" logo, and at the top right is a user profile icon labeled "Okxy". The main content area is titled "Notebook compute instance". A light blue notification box states: "CPU and GPU runtime limits have changed. You can use CPU for up to 4 hours at a time with a limit of 8 hours in a 24-hour period. You can use GPU for up to 4 hours at a time with a limit of 4 hours in a 24-hour period." Below this, the "Runtime status" is "Running". The "Runtime remaining" section shows "Session: 3 h 43 m" and "Today: 7 h 42 m". The "Compute type" section has radio buttons for "CPU" (selected) and "GPU". A red "Stop runtime" button is located to the right. At the bottom, the "Notebook preview" section shows the file "Access_AWS_from_Studio_Lab_Deployment.ipynb" and a purple "Copy to project" button.

amazon SageMaker Studio Lab

Okxy

Notebook compute instance

CPU and GPU runtime limits have changed.
You can use CPU for up to 4 hours at a time with a limit of 8 hours in a 24-hour period.
You can use GPU for up to 4 hours at a time with a limit of 4 hours in a 24-hour period.

Runtime status
Running

Runtime remaining ?
Session: 3 h 43 m
Today: 7 h 42 m

Compute type ?
☒ CPU ☐ GPU

Stop runtime

Notebook preview
Access_AWS_from_Studio_Lab_Deployment.ipynb

Copy to project

Amazon Sagemaker

Veamos el entorno [SageMaker Studio Lab](#)



Hugging Face



[Hugging Face – The AI community building the future.](#)

[Hugging Face - Documentation](#)

 [Transformers Notebooks](#)


- Hugging Face fue fundada en 2016 por Clément Delangue, Julien Chaumond y Thomas Wolf.
- Comenzó como un chatbot dirigido a adolescentes.
- Se transformó en el repositorio de IA de código abierto más completo, conocido como el "GitHub de la IA".
- Democratizó el acceso a modelos y datasets de IA, facilitando el trabajo de investigadores, empresas y desarrolladores.
- Alberga más de 1M modelos y ≈250,000 datasets en áreas como NLP, visión por computadora, vision transformers.


[Hugging Face on Google Cloud](#)


[Hugging Face on Amazon SageMaker](#)

[Hugging Face on Azure – Huggingface Transformers | Microsoft Azure](#)

Hugging Face - Hub

 **Hugging Face**

Models Datasets Spaces Posts Docs Solutions Pricing 



Create a new model repository

A repository contains all model files, including the revision history.

Owner


Model name


OksanaBok / Test

License

License

Base template

☐  **Public**
Anyone on the internet can see this model. Only you (personal model) or members of your organization (organization model) can commit.


☒  **Private**
Only you (personal model) or members of your organization (organization model) can see and commit to this model.


Once your model is created, you can upload your files using the web interface or git.



Create model


[Hugging Face Hub documentation](#)

Hugging Face - Hub


 **Hugging Face**







[Models](#) [Datasets](#) [Spaces](#) [Posts](#) [Docs](#) [Solutions](#) [Pricing](#) 


 **OksanaBok** **Test**  private

[Model card](#) [Files and versions](#) [Community](#) [Settings](#) 

Test/

Metadata UI 

 license + Add License	 datasets + Add Datasets
 language + Add Languages	 metrics + Add Metrics
 base_model + Add Base Model	new_version + Add New Version
 pipeline_tag <input type="text" value="Auto-detected"/>	library_name + Add Library
tags + Add Tags	Eval Results View doc

Edit **Preview** 

NEW [Import model card template](#)


1

☒ Commit directly to the main branch

☐ Open as a pull request to the main branch

[Hugging Face Hub documentation](#)

Hugging Face - Hub

 **Hugging Face**

Search models, datasets, users...

Models Datasets Spaces Posts Docs Solutions Pricing



 OksanaBok / **Test** private

Image Classification

Model card **Files and versions** Community Settings

main Test 1 contributor History: 7 commits + Add file










 OksanaBok

Upload testcode.ipynb

b656667


VERIFIED

less than a minute ago

 .gitattributes	 Safe	1.52 kB		initial commit	15 minutes ago
 README.md	 Safe	104 Bytes		Create README.md	11 minutes ago
 testcode.ipynb	 Safe	1.86 kB		Upload testcode.ipynb	less than a minute ago

[Hugging Face Hub documentation](#)

Hugging Face - Hub

 **Hugging Face**

Search models, datasets, users...

Models Datasets Spaces Posts Docs Solutions Pricing

OksanaBok / **Test** private

Image Classification

Model card Files and versions Community Settings

main Test / testcode.ipynb

OksanaBok Upload testcode.ipynb b656667 VERIFIED 4 minutes ago

</> raw Copy download link history blame edit delete Safe 1.86 kB

In []:

```
from transformers import AutoImageProcessor, ViTForImageClassification
import torch
import matplotlib.pyplot as plt
from PIL import Image
import requests
import io


# Download an image with cute cats
url = "https://huggingface.co/datasets/huggingface/documentation-images/resolve/main/coco_sample.png"
image_data = requests.get(url, stream=True).raw
image = Image.open(image_data)

image_processor = AutoImageProcessor.from_pretrained("google/vit-base-patch16-224")
model = ViTForImageClassification.from_pretrained("google/vit-base-patch16-224")
```

Open in Colab


[Hugging Face Hub documentation](#)

Hugging Face - Hub

 **Hugging Face**

Search models, datasets, users...

Models Datasets Spaces Posts Docs Solutions Pricing

**Oksana Bokhonok**
OksanaBok

Profile

Account

Authentication

Organizations

Billing

Access Tokens

SSH and GPG Keys

Webhooks

Papers

Notifications

Local Apps and Hardware NEW

Gated Repositories

Content Preferences

Connected Apps

Theme

< Create new Access Token

Token type

Fine-grained Read Write

This cannot be changed after token creation.

Token name

Token name

User permissions (OksanaBok)

Repositories

☐ Read access to contents of all repos under your personal namespace

☐ Read access to contents of all public gated repos you can access

☐ Write access to contents/settings of all repos under your personal namespace

Webhooks

☐ Access webhooks data

☐ Create and manage webhooks

Discussions & Posts

☐ Interact with discussions / Open PRs on repos under your personal namespace

☐ Interact with discussions / Open PRs on external repos

☐ Interact with posts

Inference

☐ Make calls to the serverless Inference API

☐ Make calls to Inference Endpoints

☐ Manage Inference Endpoints

Collections

☐ Read access to all collections under your personal namespace

☐ Write access to all collections under your personal namespace

Billing

☐ Read access to your billing usage and know if a payment method is set

[Hugging Face Hub documentation](#)

Hugging Face - Hub

Veamos el entorno [Hugging Face Hub documentation](#)



Hugging Face - Instalación





Installation

!pip install transformers datasets huggingface_hub

- **transformers:** para trabajar con modelos pre entrenados.
- **datasets:** para cargar y procesar datasets.
- **huggingface_hub:** para gestionar y autenticar en Hugging Face Hub.
 - <https://huggingface.co/settings/tokens>

```
from huggingface_hub import login  
login(token="tu_token_aqui")
```

Hugging Face - Clases

Proceso básico	Ejemplo de clasificación: image_classification.ipynb - Colab Ejemplo: CEIA-ViT/TrabajosPracticos/TP3/TP3.ipynb at main · FIUBA-Posgrado-Inteligencia-Artificial/CEIA-ViT
Carga de datos Quickstart , quickstart.ipynb - Colab	<pre>from datasets import load_dataset ds = load_dataset('beans')</pre>
Procesamiento de imágenes, basado en tipo de modelo Image Processor Preprocess	<pre>from transformers import ViTImageProcessor model_name_or_path = 'google/vit-base-patch16-224-in21k' processor = ViTImageProcessor.from_pretrained(model_name_or_path)</pre>
Carga de modelo  Transformers	<pre>from transformers import ViTForImageClassification labels = ds['train'].features['labels'].names model = ViTForImageClassification.from_pretrained(model_name_or_path,)</pre>
Entrenamiento/Finetuning Trainer , Fine-tune a pretrained model , Fine-Tune ViT for Image Classification with  Transformers Fine-tune a pretrained model	<pre>from transformers import Trainer trainer = Trainer(model=model, args=training_args, data_collator=collate_fn,)</pre>
Evaluación  Evaluate	 Evaluate

Hugging Face. Ejemplos de modelos

Multimodal. Generación de captions (texto a partir de imagen)	nlconnect/vit-gpt2-image-captioning · Hugging Face
Multimodal. Generación de captions (texto a partir de imagen)	Salesforce/blip-image-captioning-large · Hugging Face
Multimodal. Clasificación y búsqueda de imágenes basado en texto	openai/clip-vit-large-patch14 · Hugging Face
Clasificación y detección de objetos en imágenes	microsoft/swin-base-patch4-window12-384 · Hugging Face
Clasificación de imágenes	microsoft/cvt-13 · Hugging Face
Clasificación de imágenes en dispositivos móviles	apple/mobilevit-small · Hugging Face
Aprendizaje de representaciones de imágenes sin etiquetas (clasificación y extracción de características)	facebook/dinov2-base · Hugging Face
Multimodal: Tareas de comprensión multimodal (texto e imagen)	facebook/flava-full · Hugging Face

Desafío de 30 min

Vamos a organizar grupos a través de Google Meet.

Tarea de Cada Grupo:

Crear una notebook que realice lo siguiente:

- Leer una imagen.
- Cargar un modelo pre entrenado de huggingface.
- Realizar una inferencia utilizando ese modelo.

¡Buena suerte con la tarea!

Ejemplo de fine-tuning

[CEIA-ViT/TrabajosPracticos/TP3/ViT_fine_tuning.ipynb at main · FIUBA-Posgrado-Inteligencia-Artificial/CEIA-ViT](#)

Bibliografía

-

Preguntas?