



Image Classification Brein Challenge 2020

Clasificación de Imágenes para la mejora del
proceso de inventarios en una empresa retail

Arian Gallardo
Febrero 2020



Motivación

- **Para la presentación:**
 - Presentar resultados de la automatización del proceso de clasificación de imágenes.
 - Presentar la aplicación de Aprendizaje Profundo (Deep Learning) como solución satisfactoria a problemáticas de este tipo y sus distintas aplicaciones.
- **Para la realización de este reto:**
 - Descubrir nuevos métodos y aplicaciones de Deep Learning y poder entender su escalabilidad a problemas en nuevos contextos.
 - Divertirme aprendiendo cosas interesantes de Computer Science :)

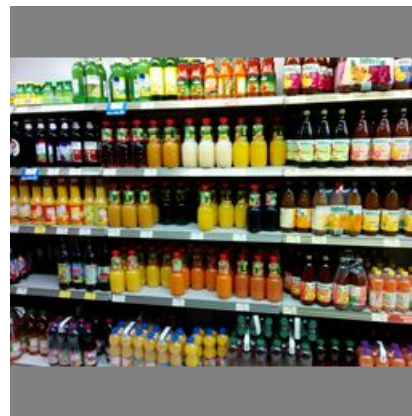
Visualización de los datos: 3215 imágenes



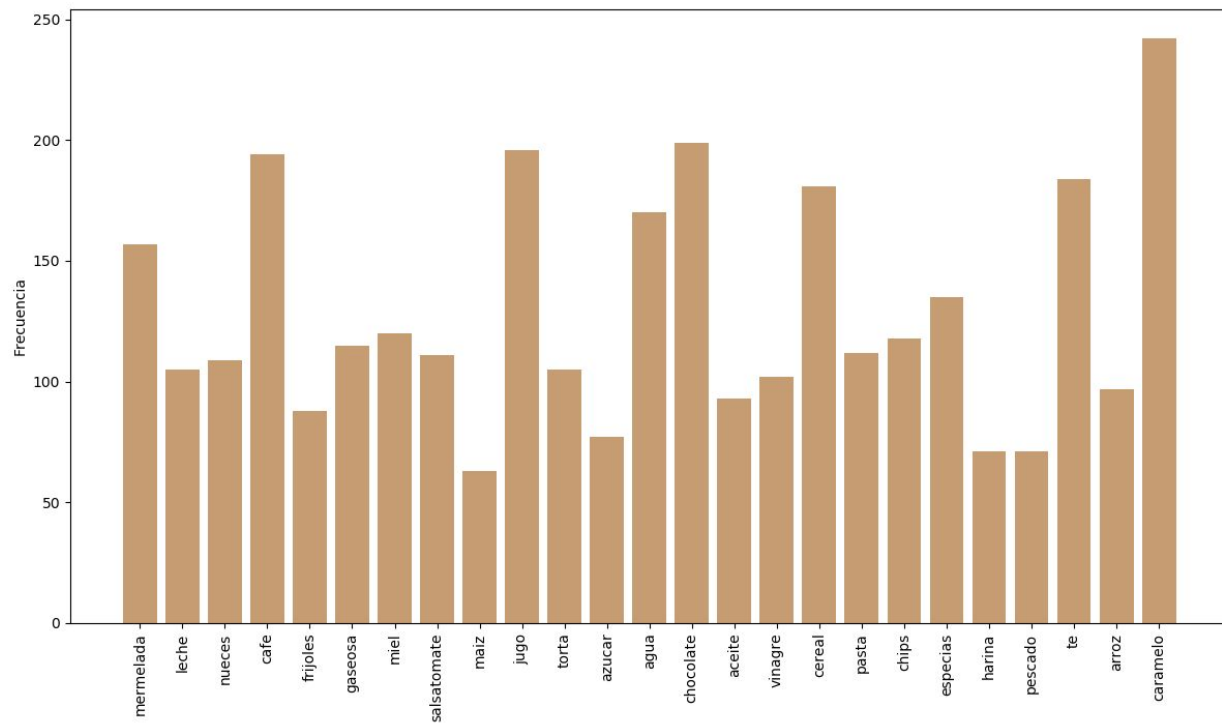
salsatomate-35.png



chips-105.png



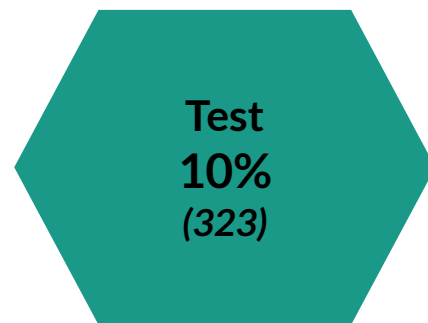
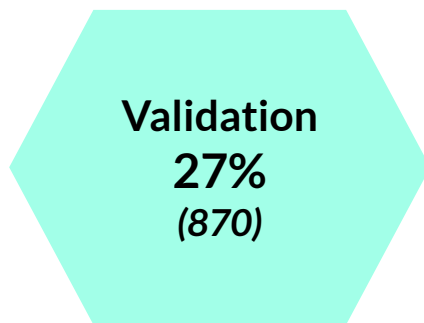
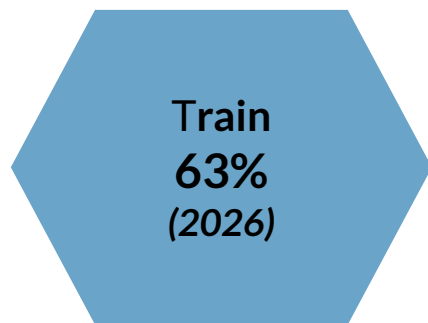
jugo-31.png



Distribución del dataset presentado para este reto



Time to split: Distribución train, val, test



Se tiene poca data, por lo cual es preferible distribuirla de esta manera.



Preprocesamiento del Train Dataset

- Normalización
- Redimensionamiento a (224, 224)
- Aún se sigue teniendo el problema de la escasez de datos!

Poca data == Data augmentation!

- Es necesario utilizar técnicas de data augmentation para compensar la escasez de la misma.



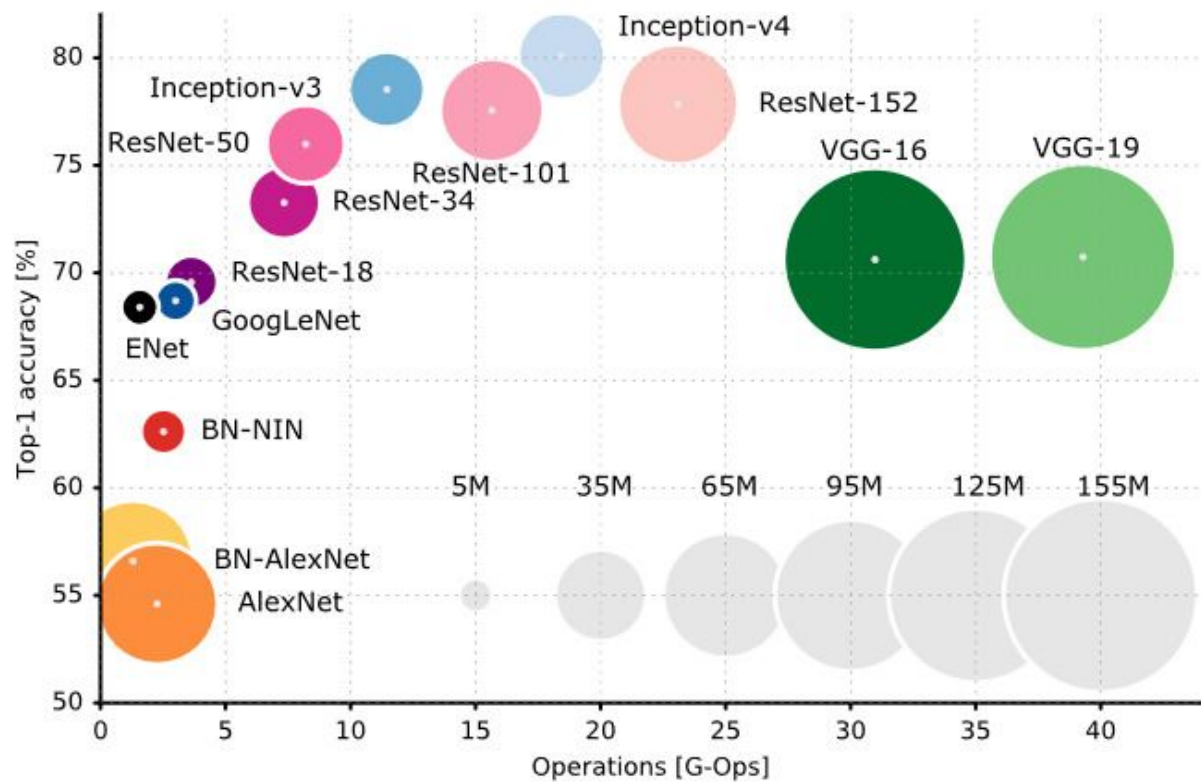
especias-100-rotated.png



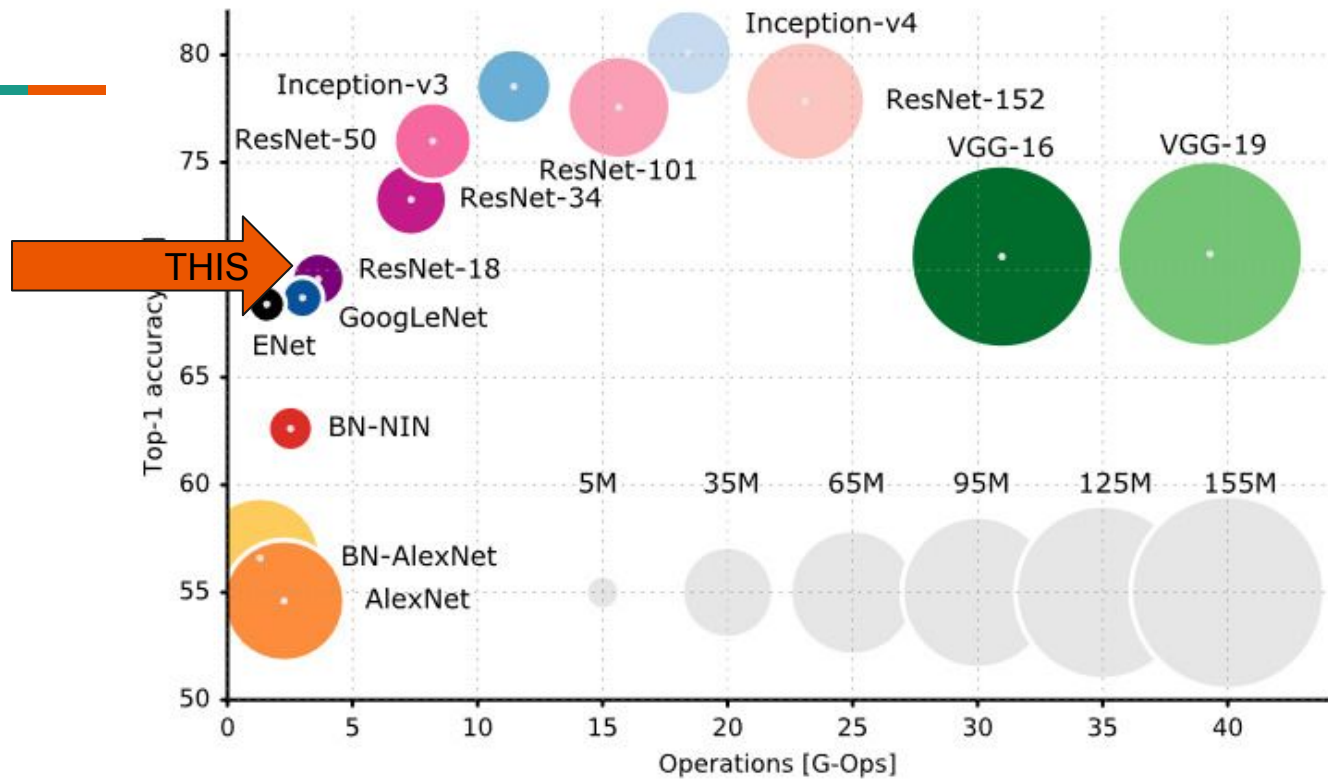
torta-37-mirror.png

**Mejor alternativa:
Convolutional Neural
Networks (CNNs)**

—



From "An Analysis of Deep Neural Network Models for Practical Applications" - Alfredo Canziani et al.



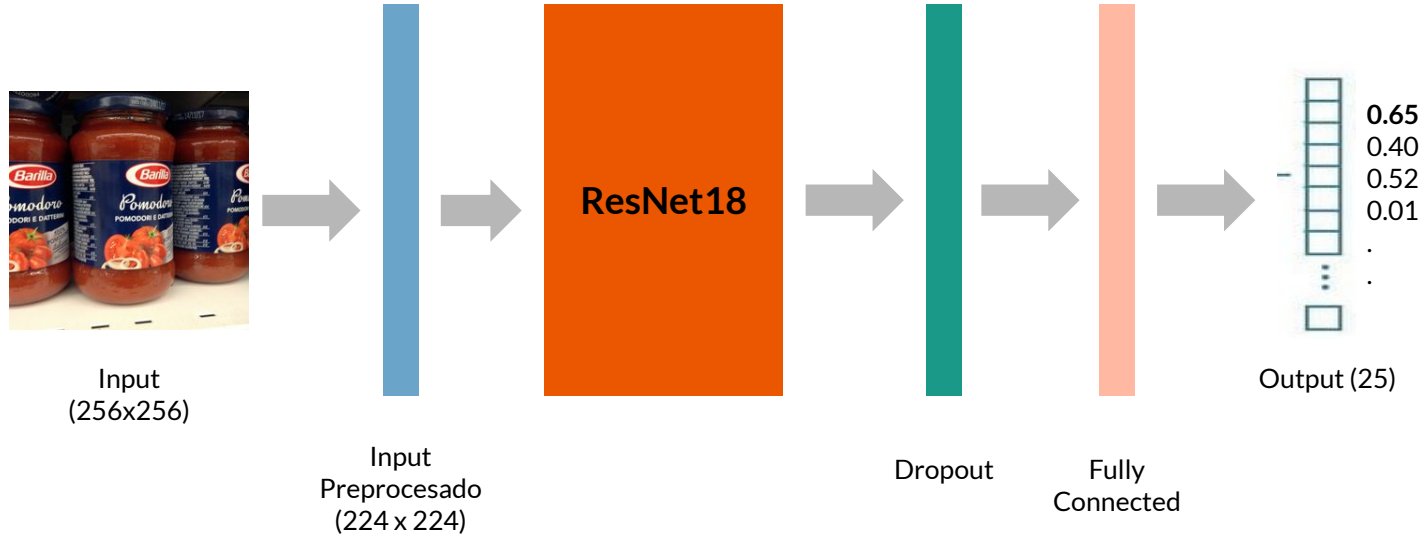
From "An Analysis of Deep Neural Network Models for Practical Applications" - Alfredo Canziani et al.



¿Por qué ResNet-18?

- Cantidad mediana de parámetros
- Entrenado sobre imágenes naturales (ImageNet)
- Disponibilidad de hardware
- La red con mejor performance que cumple todos los anteriores requisitos

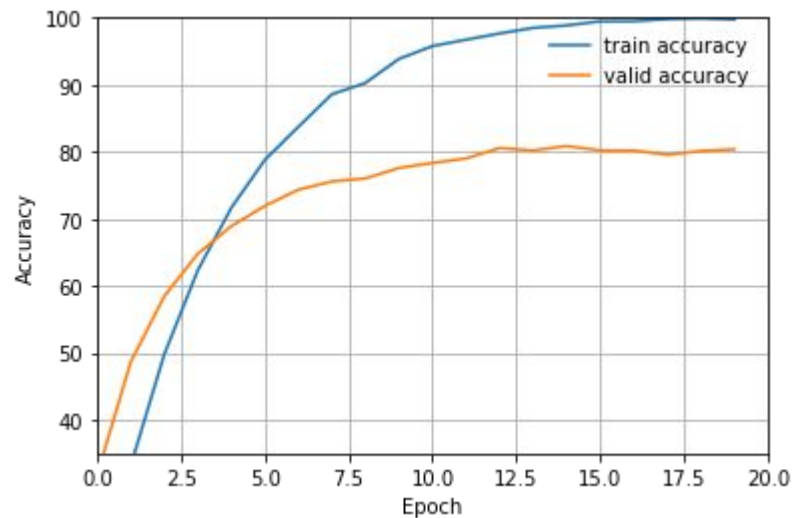
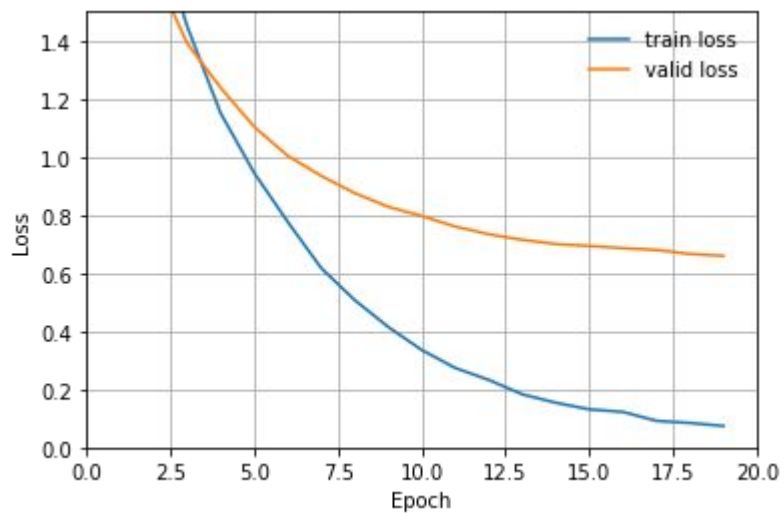
Arquitectura de la red:





Entrenamiento de la red

- Se precargan los pesos de ResNet-18 entrenada en ImageNet
- Se escogió un learning rate bajo para realizar un ajuste pequeño a los parámetros ya entrenados
- Loss: Negative Log Likelihood loss
- Optimizers: Adam
- Épocas: 20
- Se escoge el modelo con el menor validation_loss entre todas las épocas



Train loss vs Validation loss | Train accuracy vs Validation accuracy

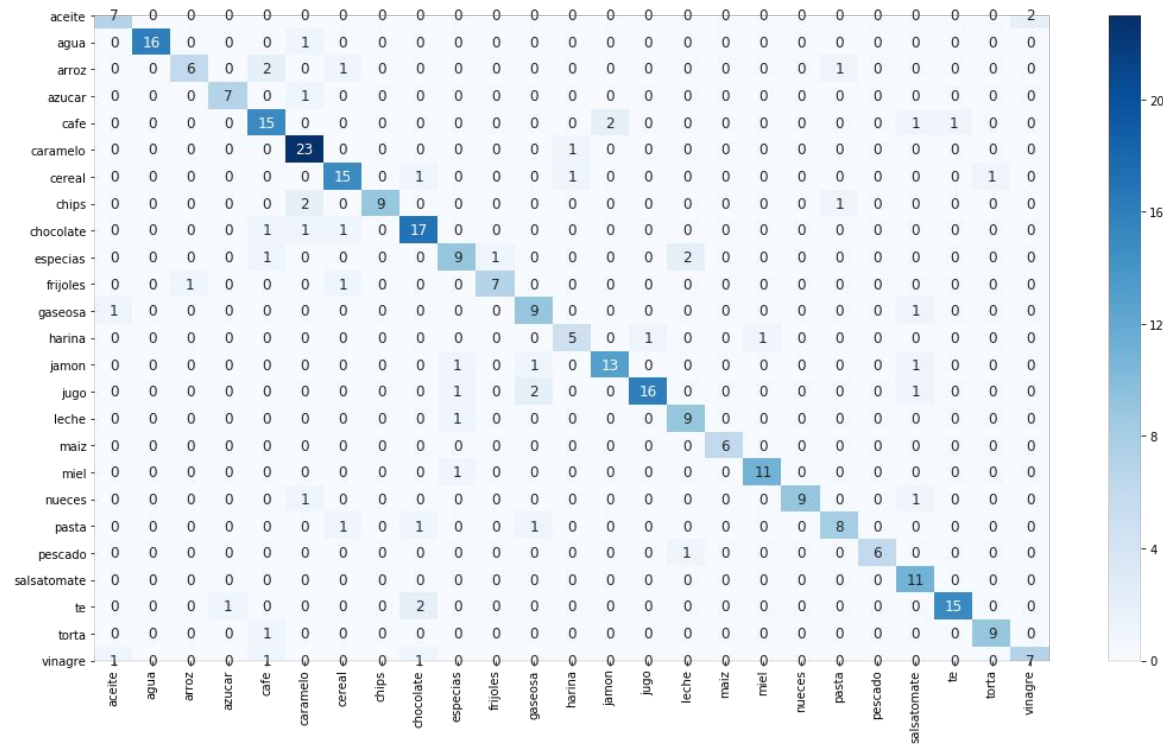
80.38% val_acc

Validation loss: 0.6595

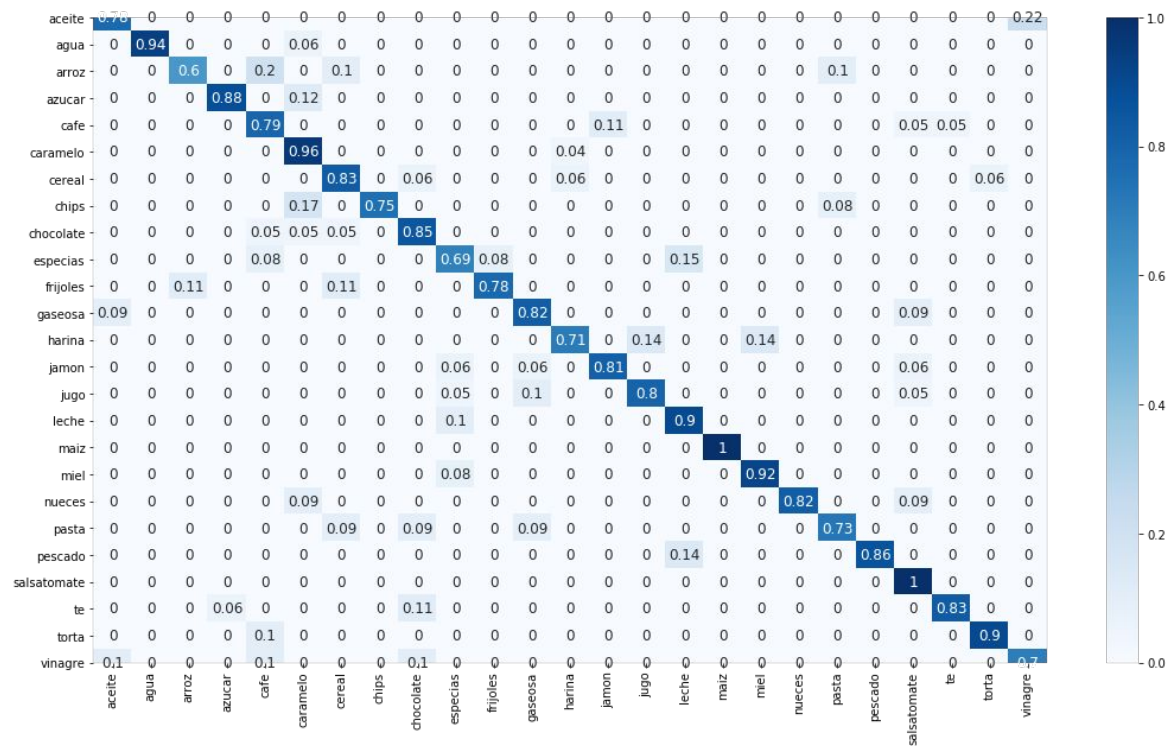
Alcanzado en la época 20, en una validación sobre el 27% del dataset.

83% test_acc

Testing realizado sobre el 10% del dataset.



Matriz de confusión



Matriz de confusión (normalizada)

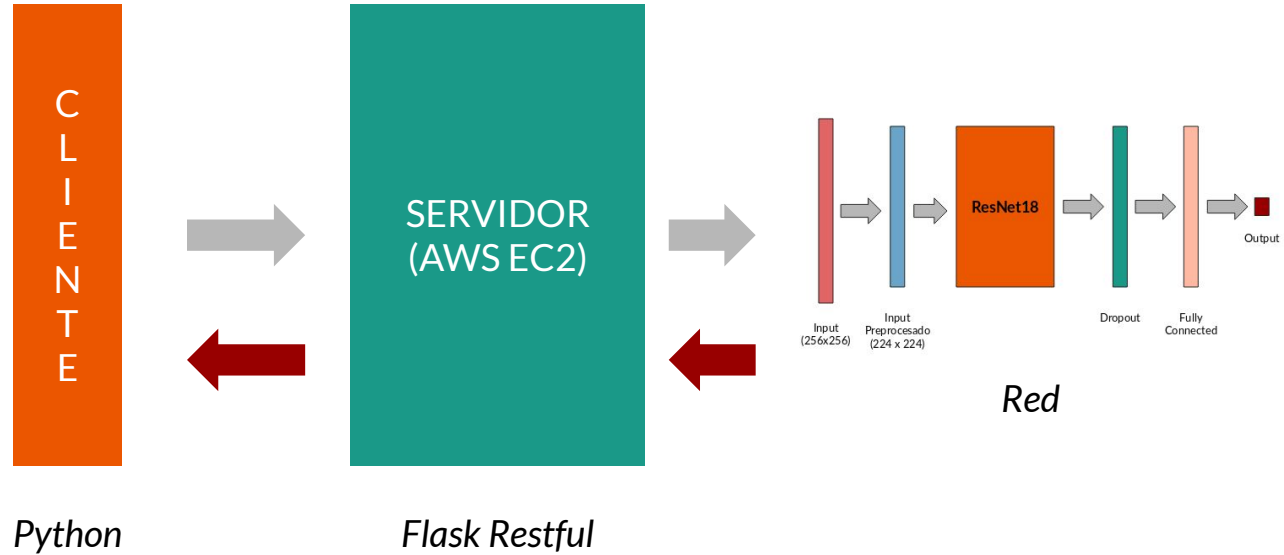


Reflexión acerca de la data

- Poca data! Se requiere de redes buenas en ello y técnicas de data augmentation
- Features muy parecidos entre algunas clases: se predice 22% de la clase aceite como vinagre
- Los bordes oscuros pueden obstruir en la clasificación
- Aún así, la clasificación ha generado resultados prometedores

Despliegue del clasificador (API)

Arquitectura del API





Software Usado

PYTORCH





Conclusiones

- Frente a escasez de datos se requieren necesariamente técnicas de data augmentation
- Realizar Transfer Learning sobre datasets como este es muy certero, pues se cuenta con una variedad gigante de models pre entrenados con imágenes parecidas
- ResNet-18 es suficiente frente a este tipo de problemas, redes más potentes podrían hacer overfitting (por la cantidad de parámetros)
- El despliegue del API se hace sencillo gracias a la variedad de librerías que Python puede usar



Aplicaciones de negocio de este reto

- Con la ayuda de más técnicas de visión computacional y machine learning, se puede obtener una herramienta para reducir tiempos de inventariado.
- Se puede controlar fácilmente la disponibilidad de un producto, ya sea en un almacén, supermercado o inventario, con lo cual se puede elaborar un plan estratégico de abastecimiento de stock.
- Podemos generar mayor accesibilidad a productos en tiendas, por ejemplo, desarrollando un sistema audiovisual para personas que sufren de deficiencias visuales.

Gracias!

Thank you!

Obrigado!

धन्यवाद!

谢谢你!

