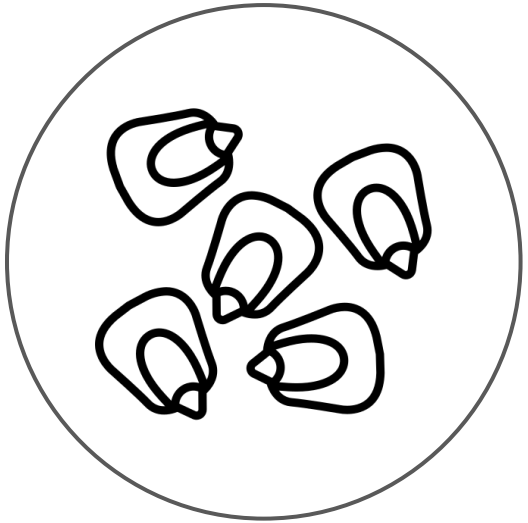

PUCP

IA PUCP - Diplomatura de Desarrollo de Aplicaciones de Inteligencia Artificial

Visión Computacional
Presentación Final

Profesor: Analy Alfaro
Alain Alejo



Integrantes

- Diego De Lama

- Erika Tello

Tema:

Clasificación de Semillas de Maíz con Redes Neuronales Convolucionales

Agenda

1. Problemática
2. Propuesta de Solución con Visión Computacional
    2.1. Tratamiento del dataset
    2.2. Clasificación de imágenes
    2.3.. Detección de objetos
3. Conclusiones y Recomendaciones

1. Problemática

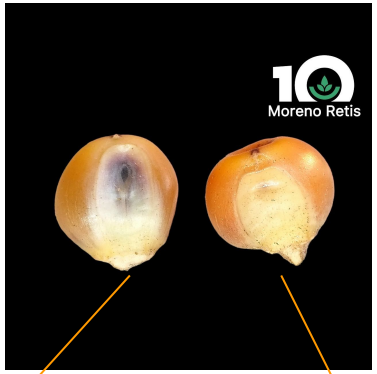
Contexto de la problemática

- Empresa dedicada a la producción de semillas de maíz y girasol.

Población

Mazorcas de inducción

Granos de maíz (mazorcas desgranadas)

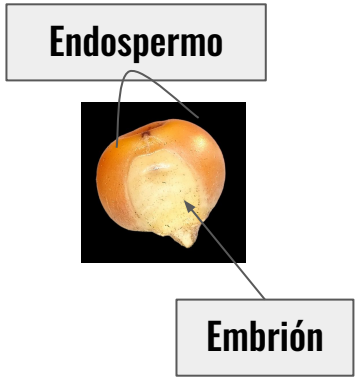


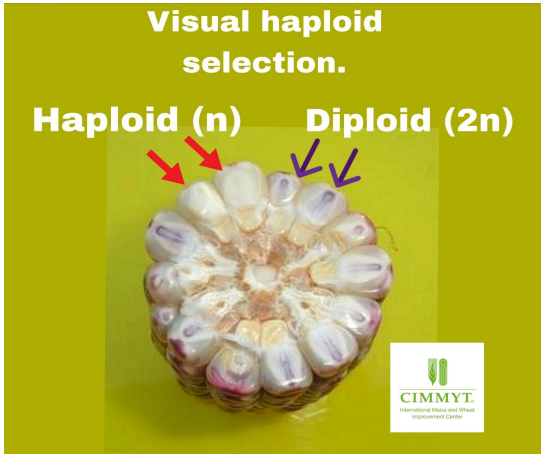
Diploides

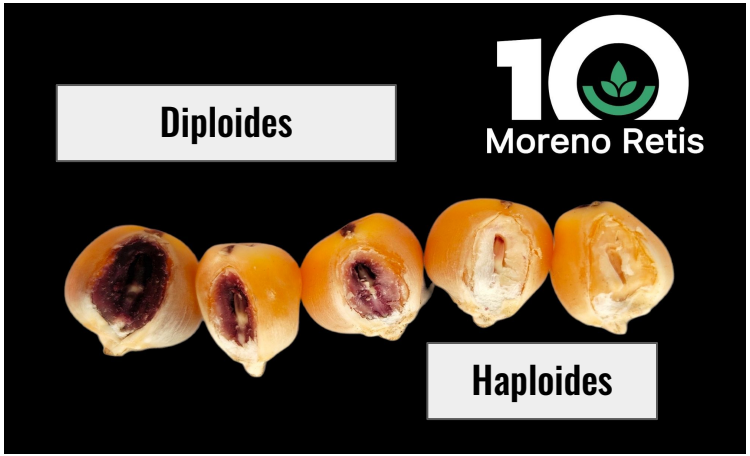
Haploides

¿Cómo se diferencian los haploides de los diploides?

Diferencia en la marcación del embrión y el endospermo del grano de maíz. Un grano haploide (útil para los próximos procesos), debe tener una marcación en el endospermo y no debe tener marcación en el embrión.

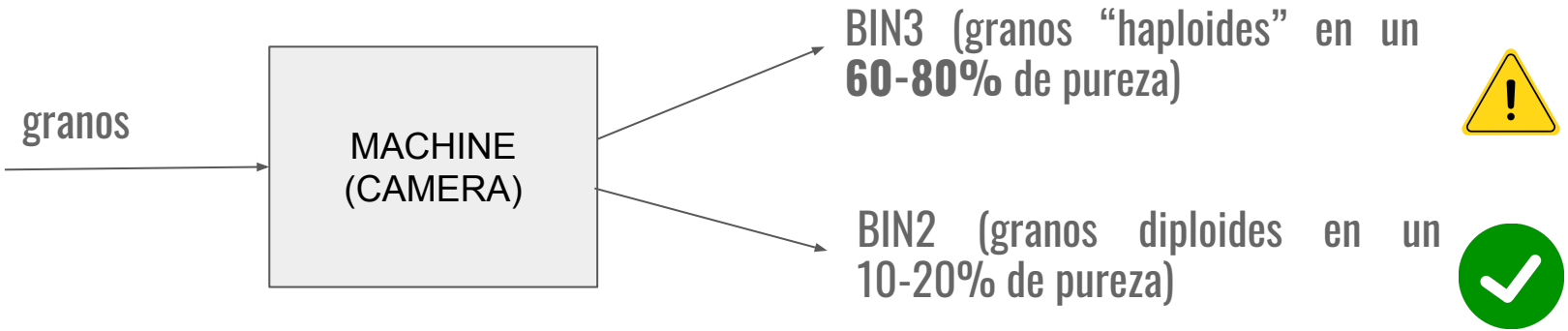






¿Cuál es el problema?

- Una población de 3 kilogramos puede tener cerca de 10,000 granos. De los cuales solo se desea seleccionar los granos haploides, ya que serán útiles para los próximos pasos del proceso.
- Los diploides son considerados descartes y están presentes en un 80%-88% del total de una población.



¿Cuál es el problema?

- Ese porcentaje de pureza del BIN3, ocasiona que en próximas etapas se siembre semilla que debe ser descartada, lo que ocasiona incremento de costos y desperdicios.
- Las diferencias entre haploides y diploides son visuales → Se puede proponer un modelo de visión computacional.

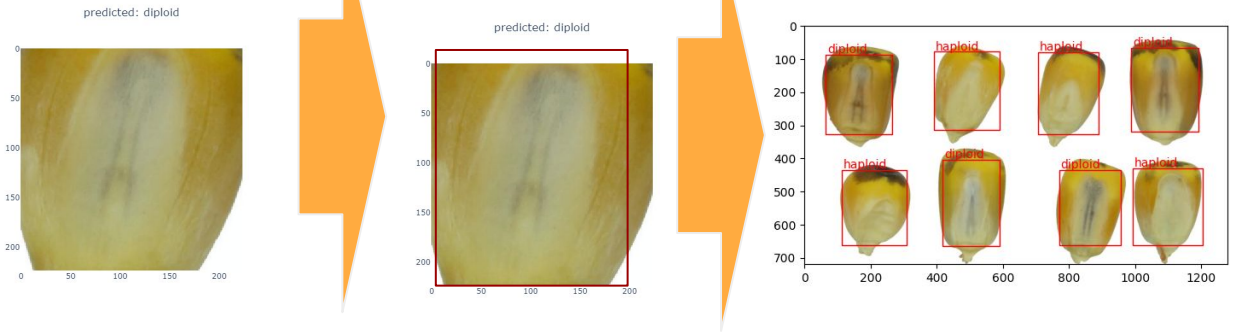
2. Propuesta de Solución con Visión Computacional

¿Cómo procederemos?

Clasificación

Clasificación
+
Localización

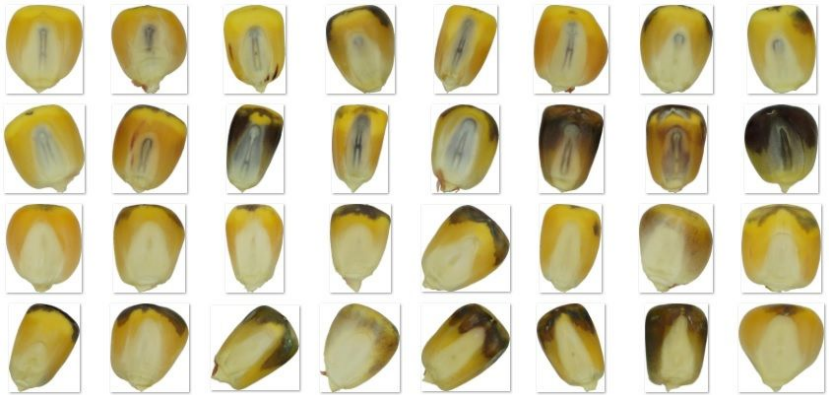
Detección de objetos



2.1. Tratamiento del Dataset

Fuente de datos

- Haploid and Diploid Maize Seeds Dataset: incluye 3,000 imágenes etiquetadas.
  - 1,230 correspondientes a haploides
  - 1,770 correspondientes a diploides.



- El dataset se dividió en train_dataset (80%) y test_dataset (20%).

https://www.rovile.org/datasets/haploid-and-diploid-maize-seeds-dataset/

Transformaciones a las imágenes

Se aplicaron transformaciones a las imágenes del dataset.

- Resize → 255
- CenterCrop → 224
- Normalización

2.2. Clasificación de imágenes

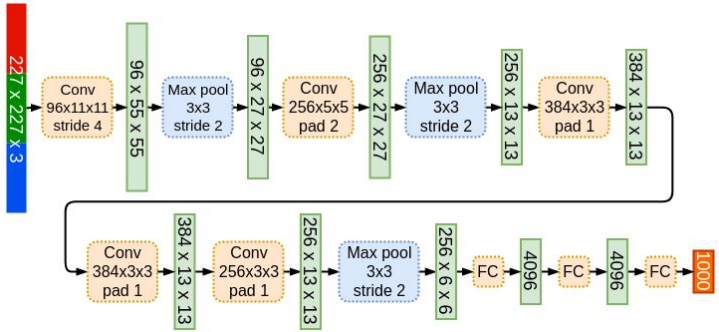
Redes neuronales evaluadas

Técnica utilizada - Transfer learning

- Se evalúa 5 modelos de clasificación preentrenados
- Se freezea los parámetros de 4 de los 5 modelos (no se realizan cambios en sus parámetros), excepto en la última capa.
- Se modifica la última capa
- Se entrenó 25 épocas

CNN con AlexNet

Arquitectura



Cambios en el modelo

Modificación en la última capa, capa Linear de:
in_features=4096
out_features=2 (correspondiente a las 2 categorías: haploide y diploide)

Gráfico de Pérdidas

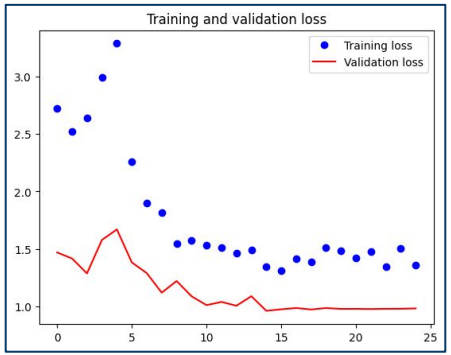
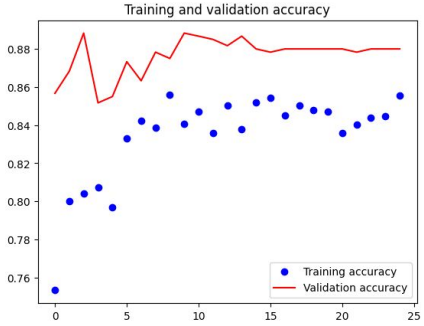
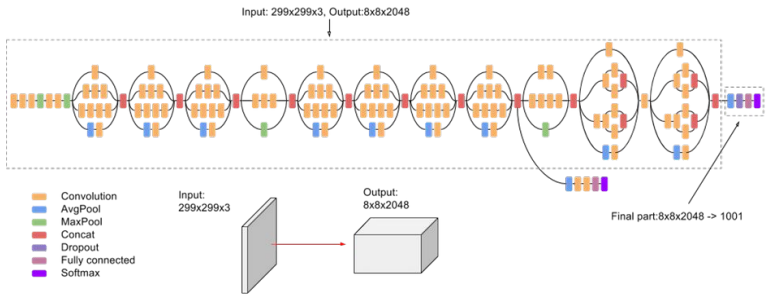


Gráfico de Precisión



CNN con Googlenet

Arquitectura



Cambios en el modelo

Modificación en la última capa, capa Linear de:
in_features=1024
out_features=2 (correspondiente a las 2 categorías: haploide y diploide)

Gráfico de Pérdidas

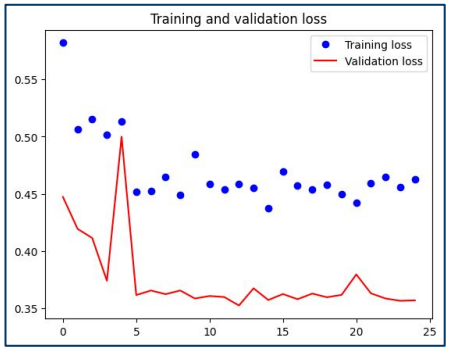
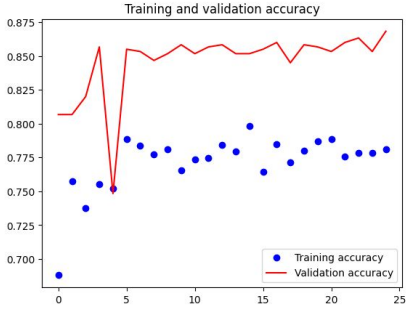
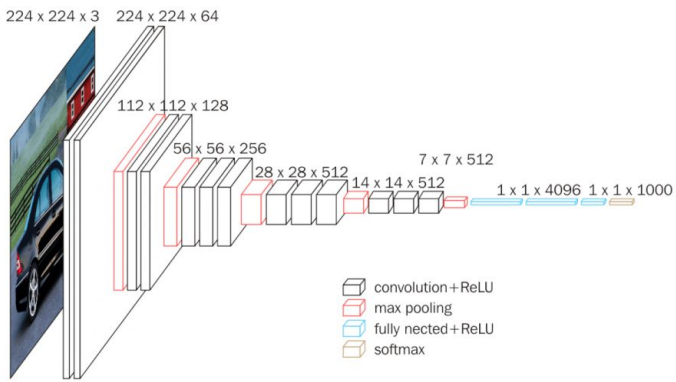


Gráfico de Precisión



CNN con VGG Net

Arquitectura



Cambios en el modelo

Modificación en la última capa, capa Linear de:
in_features=4096
out_features=2 (correspondiente a las 2 categorías: haploide y diploide)

Gráfico de Pérdidas

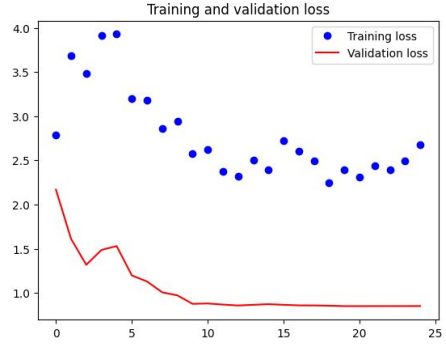
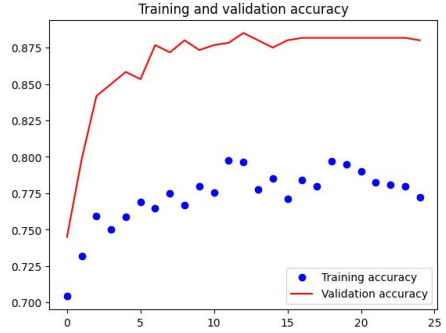


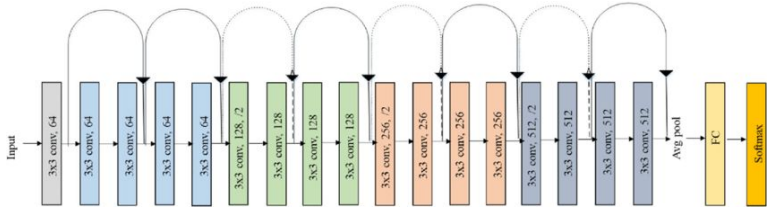
Gráfico de Precisión



Transfer Learning con ResNet 18

Arquitectura

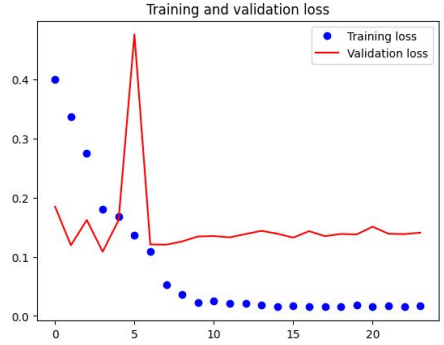
Variación de residual network, es una red neuronal convolucional con 18 capas densas.



Cambios en el modelo

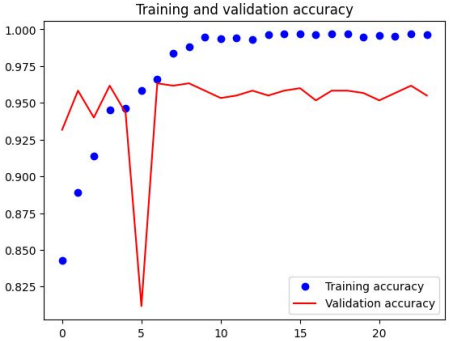
Modificación en la última capa, capa Linear de:
in_features=512
out_features=2 (correspondiente a las 2 categorías: haploide y diploide)

Gráfico de Pérdidas



0.1243

Gráfico de Precisión

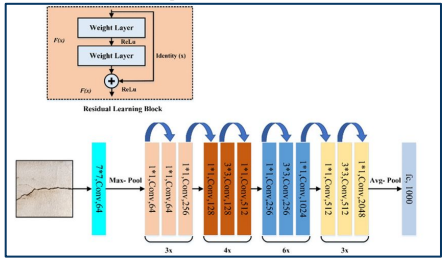


0.9500

Transfer Learning con ResNet 50

Arquitectura

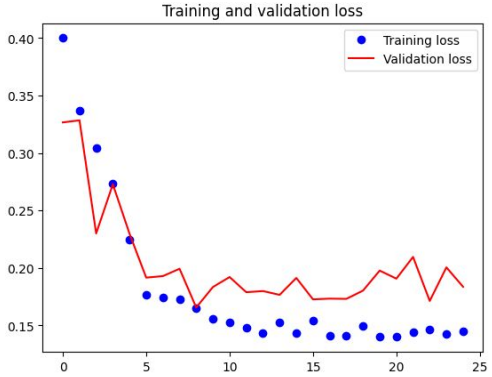
Variación de residual network, es una red neuronal convolucional con 50 capas densas.



Cambios en el modelo

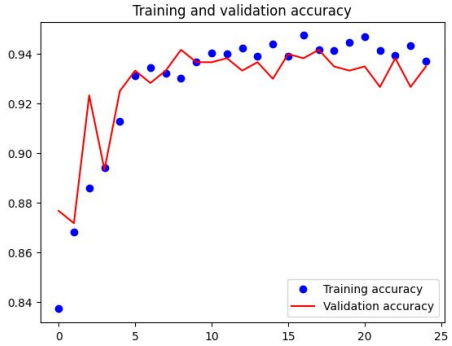
Modificación en la última capa, capa linear de:
in_features=2048
out_features=2 (correspondiente a las 2 categorías: haploide y diploide

Gráfico de Pérdidas



0.1355

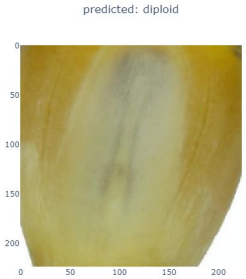
Gráfico de Precisión

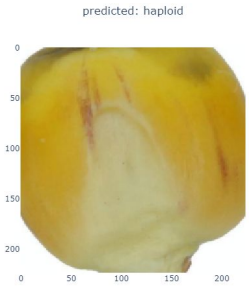


0.9533

Resumen

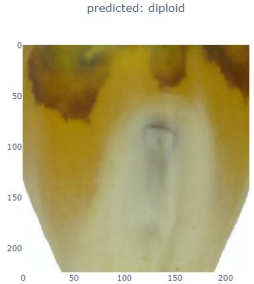
Alexnet

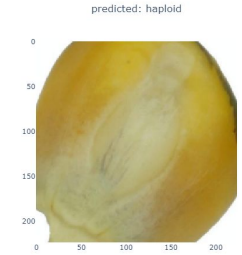




Best Val Accuracy: 88.8%
Val Loss: 1.29

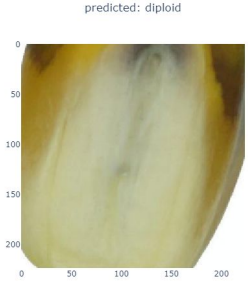
Googlenet





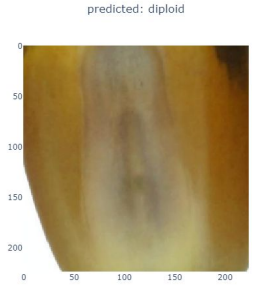
Best Val Accuracy: 85.2%
Val Loss: 0.36

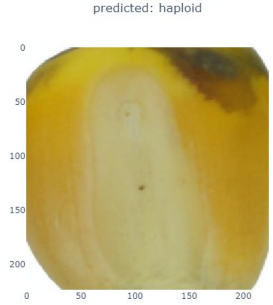
VGGnet



Best Val Accuracy: 87.52%
Val Loss: 0.87

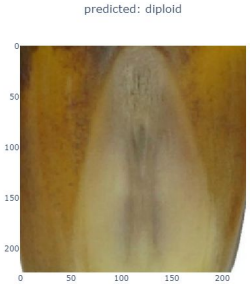
Resnet 18

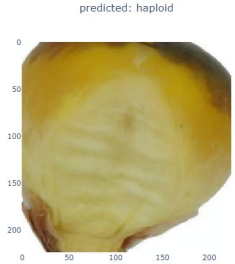




Best Val Accuracy: 95.0%
Val Loss: 0.12

Resnet 50





Best Val Accuracy: 95.3%
Val Loss: 0.14

2.2. Detección de objetos

Detección de semillas de maíz con Detecto

PUCP

Colección de imágenes

Etiquetado de las clases

Instalación de paquetes

Aumentación de imágenes

Entrenamiento

Guardado y predicción del Modelo

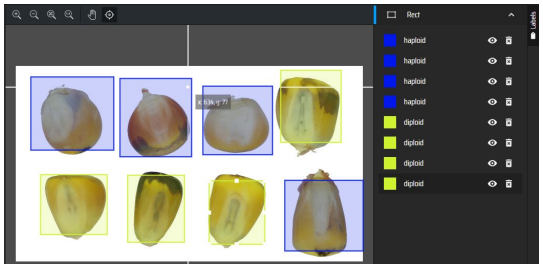
gdown

Train (20)

Test (20)

Validación (20)

MAKESENSE AI



```
custom_transforms = transforms.Compose([
transforms.ToPILImage(),
transforms.Resize(600),
transforms.RandomHorizontalFlip(0.5),
transforms.ColorJitter(saturation=0.2),
transforms.ToTensor(),
utils.normalize_transform(),
])
```

Detecto

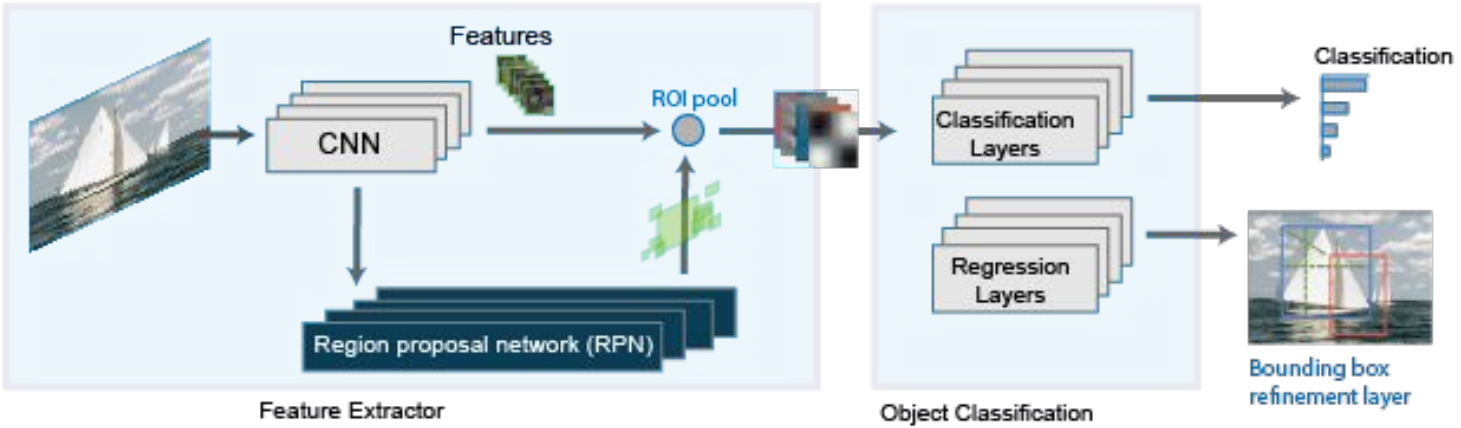
Faster R-CNN ResNet-50 FPN

```
<name>haploid</name>
<pose>Unspecified</pose>
<truncated>0</truncated>
<difficult>0</difficult>
<bndbox>
        <xmin>440</xmin>
        <ymin>431</ymin>
        <xmax>608</xmax>
        <ymax>631</ymax>
</bndbox>
</object>
<object>
```

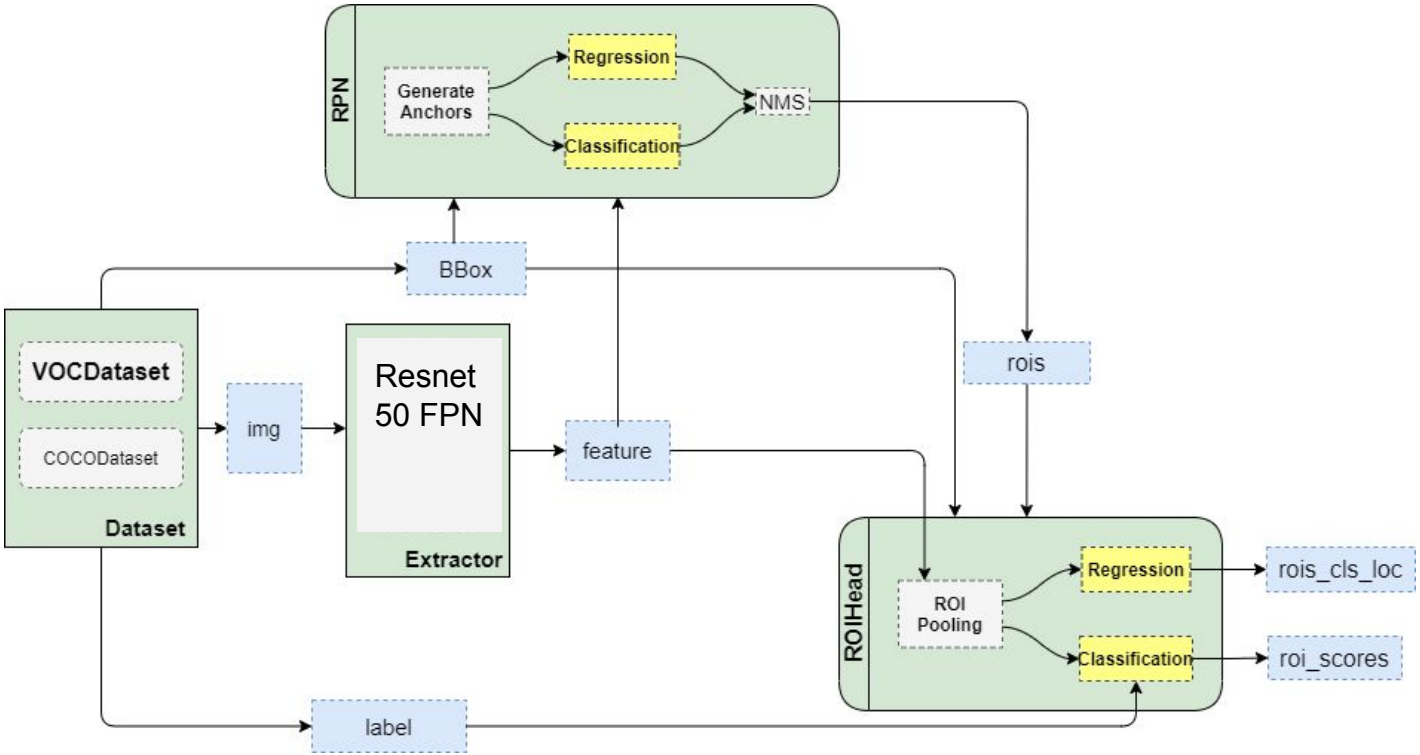
```
model=core.Model(['diploid','haploid'])
losses = model.fit(loader, Val_dataset,
epochs=25,lr_step_size=5,
learning_rate=0.001, verbose=True)
```

Arquitectura del Faster RCN Resnet 50 FPN

Faster RCN



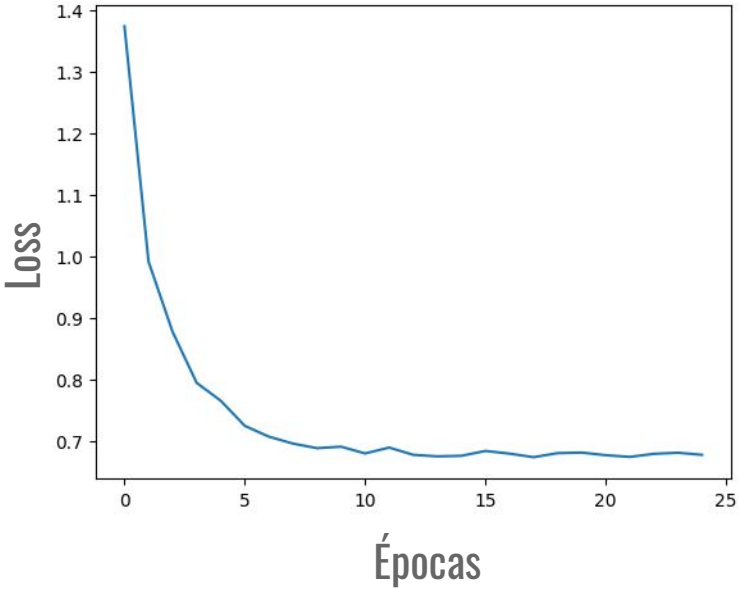
Arquitectura del Faster RCN Resnet 50 FPN



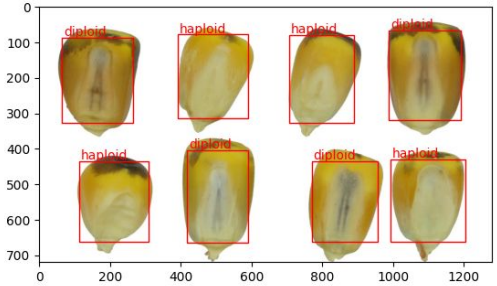
https://www.lablab.top/post/how-does-faster-r-cnn-work-part-i/

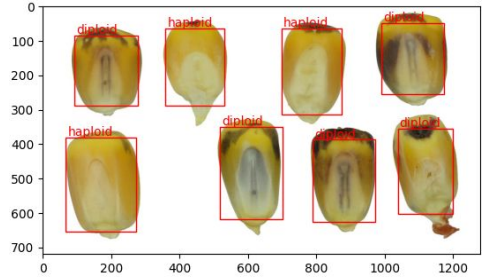
Resultados

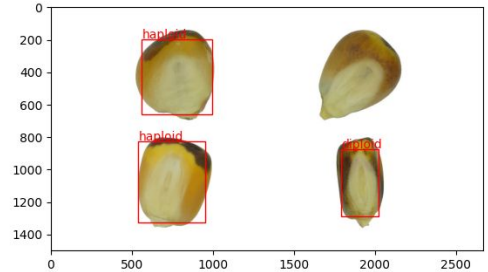
Gráfico de Pérdidas

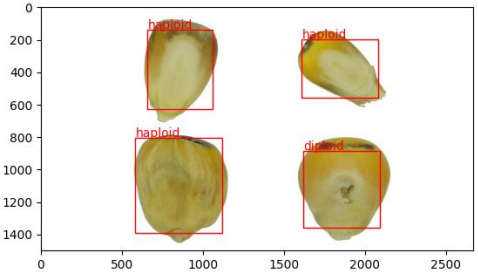


Definimos el Threshold=0.62 para los bounding boxes









3. Conclusiones y recomendaciones

Conclusiones y recomendaciones

- Para mejorar el performance de los modelos preentrenados con redes convulocionales fue necesario ‘freezear’ las capas.
- En base a la experimentación con CNN y Transfer learning, podemos concluir que el mejor modelo, entre los cinco evaluados, es ResNet50, debido a que alcanza un mayor valor de accuracy (0.95) y un menor valor de pérdida.
- Como una solución al problema propuesto, se puede mejorar el algoritmo empleado por la máquina de selección de haploides. Este que permitirá una mejor clasificación de cada grano, siendo asignado a la categoría correcta: haploide o diploide.
- Para la detección de objetos usando el paquete *Detecto* es necesario tener el etiquetado de los objetos en formato XML cuidando el tamaño de los boxes.

Referencias web

- https://subscription.packtpub.com/book/data/9781789956177/5/ch05lvl1sec13/introducing-alexnet
- https://towardsdatascience.com/review-fpn-feature-pyramid-network-object-detection-262fc7482610
- https://towardsdatascience.com/deep-learning-googlenet-explained-de8861c82765
- https://www.kaggle.com/code/blurredmachine/vggnet-16-architecture-a-complete-guide
- https://debuggercafe.com/object-detection-using-pytorch-faster-rcnn-resnet50-fpn-v2/
- https://deepsense.ai/region-of-interest-pooling-explained/
- https://blog.paperspace.com/faster-r-cnn-explained-object-detection/
- https://www.lablab.top/post/how-does-faster-r-cnn-work-part-ii/
- https://www.analyticsvidhya.com/blog/2021/06/simplest-way-to-do-object-detection-on-custom-datasets/

¡GRACIAS!