

Apellidos: Méndez Lázaro

Nombres: Luis Fernando

Fecha: 28/01/2025

Nota:

Indicaciones:

La Duración es de 40 minutos.

La evaluación consta de 10 preguntas.

Las preguntas de opción múltiple valen 1.0 punto.

- Se entrena una red con learning rate schedule que incrementa linealmente el LR en las primeras 10 épocas (warm-up), seguido de un decaimiento tipo coseno. ¿Cuál es la principal ventaja de este enfoque respecto a un step decay?
 - Evita caer en mínimos locales mediante oscilaciones controladas. ☒
 - Acelera la convergencia inicial y ajusta finamente el LR cerca de óptimos. ☒
 - Minimiza la sensibilidad a la inicialización de parámetros. ☒
 - Reduce el riesgo de overfitting al final del entrenamiento. ☒
 - Permite un mayor LR máximo que el step decay. ☒
- ¿Cuál es la ventaja ROI Align frente a ROI Pooling? mask RCNN
 - Elimina la cuantización de coordenadas mediante interpolación bilineal. ☒
 - Aumenta la resolución espacial de los features mediante upsampling. ☒
 - Utiliza una region proposal network más precisa. ☒
 - Combina información de múltiples niveles de feature pyramids. ☒
 - Aplica label smoothing a las máscaras de segmentación. ☒
- ¿Cuál de las siguientes alternativas define mejor a una representación equivariante?
 - La salida de la red no cambia ante la transformación aplicada a la entrada. ☒
 - La representación se ajusta proporcionalmente a la magnitud de la transformación aplicada, asegurando una reducción de dimensionalidad óptima. ☒
 - La representación refleja la misma transformación aplicada a la entrada, permitiendo mantener información sobre la variación estructural. ☒
 - Garantiza un incremento monótonico en la distancia euclidiana del embedding cuando la transformación es mayor. ☒
 - Una red equivariante es un Lipschitz embedding. ☒
- Una característica de emplear transposed convolution es:
 - Suavizar los bordes de los objetos. ☒
 - Generar features invariantes a traslaciones. ☒
 - Limitar la capacidad de reconstrucción a texturas simples. ☒
 - Introduce artefactos (checkerboard artifacts). ☒
 - Preserva información de detalle procesada en el Encoder. ☒
- ¿Cuál es la función de los object queries en DETR (Detection Transformer)?
 - Generar los mapas de características piramidales necesarios para la detección multi-dimensional. ☒
 - Definir los anclajes (anchors) dinámicos basados en la distribución de píxeles de entrada. ☒
 - Reemplazar por completo el concepto de Non-Maximum Suppression. ☒
 - Transformar las coordenadas de las bounding boxes a un espacio latente, reduciendo la dimensionalidad para el post-procesamiento. ☒
 - Servir como un conjunto fijo de embeddings que es procesado con la salida del encoder para predecir las bounding boxes y las clases. ☒
- ¿Cuál de estas arquitecturas garantiza equivarianza a traslaciones?
 - MLP con activaciones ReLU. ☒
 - Self-attention con positional encoding. ☒
 - CNN con max-pooling. ☒
 - Self-attention sin positional encoding. ☒
 - CNN sin max-pooling. ☒
- Un equipo de investigación entrena un modelo de Deep Learning para clasificar microorganismos (bacterias y hongos) a partir de imágenes obtenidas mediante microscopía de campo claro (light-field). El dataset cubre varias especies, distintas condiciones de cultivo y algunas variaciones de contraste y brillo para aumentar la robustez. Sin embargo, el modelo entrenado es utilizado en imágenes obtenidas por un microscopio de barrido de electrones (SEM), con características visuales muy diferentes. ¿Cuál de los siguientes enfoques sería el más adecuado para enfrentar este problema, es decir, que el modelo logre

class

out of distri

extrapolar efectivamente a imágenes muy distintas a las que fueron usadas en el entrenamiento?

a) Aplicar Data Augmentation extensivo, incluyendo transformaciones que simulen las características del microscopio SEM, como cambios en la textura y el contraste.

b) Implementar Transfer Learning utilizando un modelo pre-entrenado en un dataset de imágenes obtenidas por microscopía SEM antes de ajustarlo con el dataset actual.

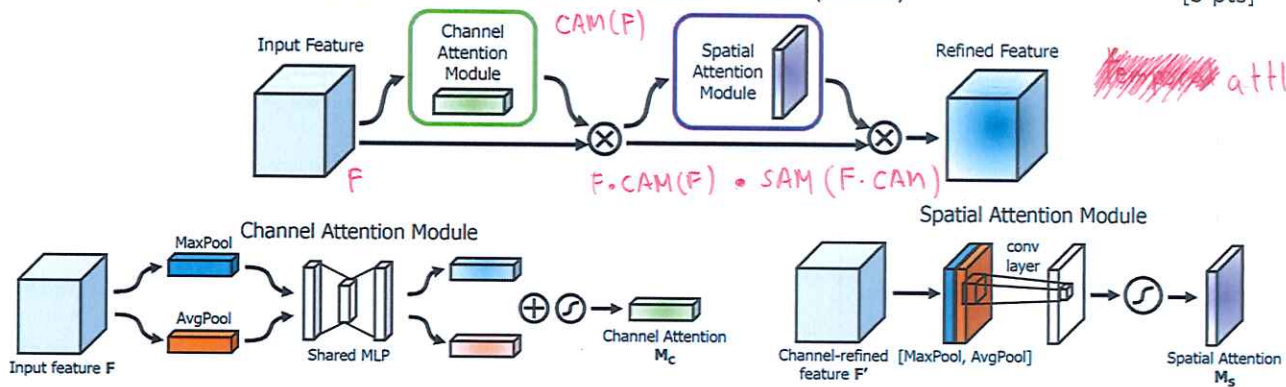
c) Emplear *Few-Shot Learning* para entrenar el modelo con un pequeño conjunto de imágenes SEM, permitiendo al modelo adaptarse rápidamente a las nuevas características visuales.

d) Utilizar Domain Adaptation mediante técnicas como Adversarial Training para alinear las distribuciones de los dominios de entrenamiento (light-field) y de prueba (SEM).

e) Congelar todas las capas del backbone y únicamente entrenar las capas superiores con el nuevo dominio SEM, minimizando el riesgo de sobreajuste

8. Implemente un módulo de Convolutional Block Attention Module (CBAM).

[5 pts]



Channel Attention M:

$\text{init}(\text{in_channels}, H, W):$

$\text{pool1} = \text{MaxPool2d}(H \times W)$

$\text{pool2} = \text{AvgPool2d}(H \times W)$

$\text{fc} = \text{Linear}(\text{in_channels}, \text{in_channels})$

$\text{forward}(F):$

$\text{out1} = \text{pool1}(F)$

$\text{out2} = \text{pool2}(F)$

flatten out1, out2

$\text{out1} = \text{fc}(\text{out1})$

$\text{out2} = \text{fc}(\text{out2})$

$\text{return } F \cdot \text{relu}(\text{out1} + \text{out2})$

Global poolings!

Spatial Attention M:

$\text{init}(\text{in_channels}, \text{out_channels}, H, W):$

$\text{pool1} = \text{MaxPool2d}(\dots)$

$\text{pool2} = \text{AvgPool2d}(\dots)$

$\text{conv} = \text{Conv2d}(\text{in_channels}, \text{out_channels})$

$\text{forward}(F):$

$\text{out1} = \text{pool1}(F')$

$\text{out2} = \text{pool2}(F')$

$\text{out3} = \text{cat}([\text{out1}, \text{out2}])$

$\text{out4} = \text{conv}(\text{out3})$

$\text{return } F \cdot \text{relu}(\text{out4})$

Concatenation!

CBAM:

$\text{init}(\text{in_channels}, \text{out_channels}, H, W):$

$\text{CAM} = \text{channelAttentionM}(\dots)$

$\text{SAM} = \text{SpatialAttentionM}(\dots)$

$\text{forward}(F):$

$\text{return } (F * \text{CAM}(F) * \text{SAM}(F * \text{CAM}(F)))$

9. Proponga un pseudocódigo que implemente RepVGG. [5 pts]

RepVGG Training:

init (in_chan, out_chan):

conv3x3 = Conv2d(in_chan, out_chan, stride=2)

conv1x1 = Conv2d(in_chan, out_chan, stride=2)

forward (x):

out1 = F.relu(conv3x3(x) + Conv1x1(x))

out2 = F.relu(conv3x3(out1) + Conv1x1(out1) + out1)

out3 = F.relu(conv3x3(out2) + Conv1x1(out2) + out2)

return F.relu(conv3x3(out3) + Conv1x1(out3) + out3)

RepVGG Inference:

init (in_chan, out_chan):

conv3x3 = # Combinar pesos
aprendidos de conv3x3 y Conv1x1
de RepVGG Training :)

forward (x):

out1 = conv3x3(x)

out1 = F.relu(out1)

out2 = F.relu(conv3x3(out1))

out2 = F.relu(conv3x3(out1))

out3 = F.relu(conv3x3(out2))

return F.relu(conv3x3(out2))

10. Sobre las curvas ROC y AP, Demuestre matemáticamente cual es la más adecuada para enfrentar datos desbalanceados, y cual no. [5 pts]

AP (precision-recall curve) para datos desbalanceados

ROC para datos balanceados (métrica Global)
de rendimiento

$$\text{precision} = TP / TP + FP$$

$$\text{recall} = TP / TP + FN$$

