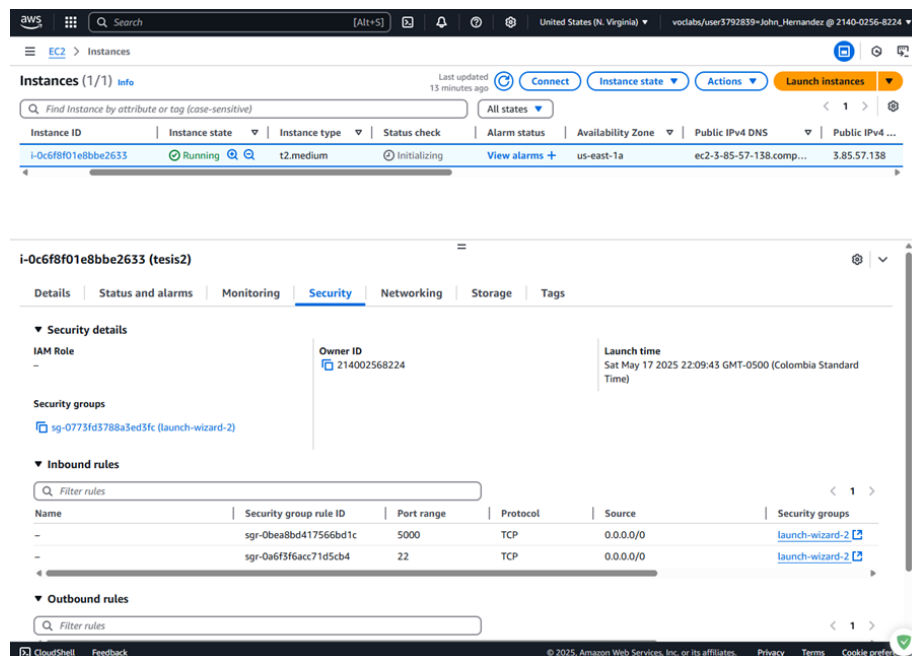


Anexo del artículo: Detección y conteo automático de mamíferos africanos en imágenes aéreas mediante aprendizaje profundo: adaptación y evaluación del modelo HerdNet

Anexo 1: Máquina virtual:

Para el trabajo de investigación se generó la configuración de una máquina virtual en AWS EC2 de tipo t2.medium. En esta, se generaron los grupos de seguridad para acceso por SSH (puerto 22) desde cualquier dirección IP y acceso al puerto 5000 (TPC) destinado a la ejecución del servicio de seguimiento de experimentos mediante MLFlow.

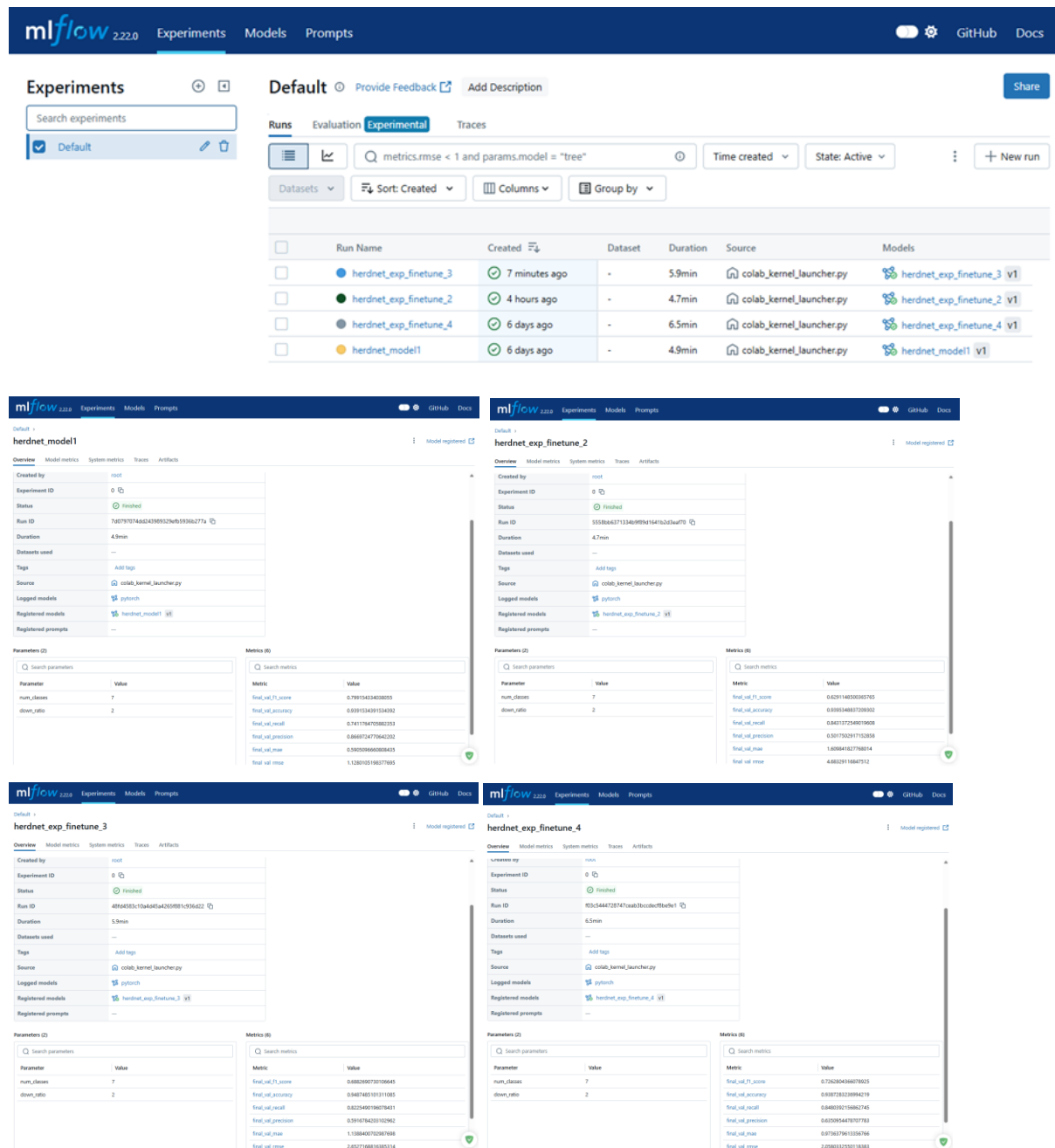


Se genera la conexión del ambiente virtual con la configuración de MLFlow:

```
ubuntu@ip-172-31-18-45: ~$ sudo apt update && sudo apt install python3-pip && sudo apt install python3-venv && python3 -m
venv mlflow-env && source mlflow-env/bin/activate
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1090 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [161 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1063 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [376 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7860 B]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [16.4 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:15 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.2 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [288 B]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [288 B]
Fetched 3167 kB in 1s (3902 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
444 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (24.0dfsg-1ubuntu1.1).
0 upgraded, 0 newly installed, 0 to remove and 444 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-venv is already the newest version (3.12.3-0ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 444 not upgraded.
(mlflow-env) ubuntu@ip-172-31-18-45:~$ pip install mlflow && mlflow server --backend-store-uri sqlite:///mlflow.db --hos
t 0.0.0.0 --port 5000
Requirement already satisfied: mlflow in ./mlflow-env/lib/python3.12/site-packages (2.22.0)
Requirement already satisfied: mlflow-skinny==2.22.0 in ./mlflow-env/lib/python3.12/site-packages (from mlflow) (2.22.0)
Requirement already satisfied: Flask<4 in ./mlflow-env/lib/python3.12/site-packages (from mlflow) (3.1.0)
Requirement already satisfied: Jinja2<4,>2.11 in ./mlflow-env/lib/python3.12/site-packages (from mlflow) (3.1.6)
Requirement already satisfied: alembic<1.10.0,<2 in ./mlflow-env/lib/python3.12/site-packages (from mlflow) (1.15.2)
Requirement already satisfied: dockere<0,>=1.0.0 in ./mlflow-env/lib/python3.12/site-packages (from mlflow) (7.1.0)
Requirement already satisfied: graphene<4 in ./mlflow-env/lib/python3.12/site-packages (from mlflow) (3.4.3)
Requirement already satisfied: gunicorn<24 in ./mlflow-env/lib/python3.12/site-packages (from mlflow) (23.0.0)
Requirement already satisfied: markdown<4,>=3.3 in ./mlflow-env/lib/python3.12/site-packages (from mlflow) (3.8)
```

Anexo 2: Evaluaciones de los modelos en MLFlow:

Luego de la conexión correspondiente con el puerto de MLFlow, se coleccionan los modelos generados desde los ambientes de Python en Colab Pro junto con sus resultados para almacenarlos en la instancia correspondiente de MLFlow:



Anexo 3: Resultados modelos por clase:

En adición a los resultados generales de los modelos, y para mayor profundización en los rendimientos de los mismos, los siguientes son los resultados de los modelos y su rendimiento por clase (tener en cuenta la asignación de cada clase con su especie, la cual es: 0 - Topis, 1 - Búfalos, 2 - Cobos, 3 - Jabalíes, 4 - Antílopes acuáticos, 5 - Elefantes, y 6- rendimiento general del modelo):

- *Experimento 1:*

	class	n	recall	precision	f1_score	confusion	mae	mse	rmse	ap
0	1	79	0.696203	0.591398	0.639535	0.098361	0.833333	1.222222	1.105542	0.594819
1	2	319	0.639498	0.829268	0.722124	0.033175	1.235849	3.820755	1.954675	0.569076
2	3	379	0.825858	0.879213	0.851701	0.071217	0.349673	0.460784	0.678811	0.770723
3	4	19	0.368421	0.350000	0.358974	0.000000	1.041667	1.791667	1.338532	0.305263
4	5	46	0.521739	0.827586	0.640000	0.272727	0.833333	1.100000	1.048809	0.452471
5	6	178	0.601124	0.835938	0.699346	0.000000	1.027027	2.297297	1.515684	0.508352
6	binary	1020	0.741176	0.866972	0.799154	0.000000	0.590510	1.272408	1.128011	0.613024

- *Experimento 2:*

	class	n	recall	precision	f1_score	confusion	mae	mse	rmse	ap
0	1	79	0.582278	0.438095	0.500000	0.352113	1.555556	11.111111	3.333333	0.356729
1	2	319	0.802508	0.354571	0.491835	0.019157	4.294643	83.616071	9.144182	0.344866
2	3	379	0.939314	0.607509	0.737824	0.011111	0.753754	3.486486	1.867214	0.789121
3	4	19	0.473684	0.214286	0.295082	0.100000	1.482759	4.379310	2.092680	0.428947
4	5	46	0.500000	0.560976	0.528736	0.425000	1.166667	1.966667	1.402379	0.328507
5	6	178	0.662921	0.541284	0.595960	0.000000	1.487179	9.205128	3.033995	0.477040
6	binary	1020	0.843137	0.501750	0.629115	0.000000	1.609842	21.933216	4.683291	0.489900

- *Experimento 3:*

	class	n	recall	precision	f1_score	confusion	mae	mse	rmse	ap
0	1	79	0.708861	0.424242	0.530806	0.188406	1.534483	7.982759	2.825378	0.434417
1	2	319	0.786834	0.470919	0.589202	0.023346	2.690909	25.272727	5.027199	0.396997
2	3	379	0.902375	0.693712	0.784404	0.031161	0.577640	1.316770	1.147506	0.751224
3	4	19	0.473684	0.272727	0.346154	0.100000	1.478261	4.782609	2.186918	0.301711
4	5	46	0.565217	0.541667	0.553191	0.297297	1.225806	2.322581	1.524002	0.376325
5	6	178	0.629213	0.625698	0.627451	0.008850	1.171053	3.986842	1.996708	0.414238
6	binary	1020	0.822549	0.591678	0.688269	0.000000	1.138840	7.036907	2.652717	0.500147

- *Experimento 4:*

	class	n	recall	precision	f1_score	confusion	mae	mse	rmse	ap
0	1	79	0.746835	0.495798	0.595960	0.169014	1.142857	2.920635	1.708987	0.532748
1	2	319	0.811912	0.507843	0.624849	0.022642	2.226891	13.588235	3.686222	0.471629
2	3	379	0.902375	0.721519	0.801876	0.039326	0.520249	0.943925	0.971558	0.769948
3	4	19	0.473684	0.300000	0.367347	0.181818	1.291667	2.958333	1.719981	0.409357
4	5	46	0.413043	0.593750	0.487179	0.500000	1.133333	1.733333	1.316561	0.290238
5	6	178	0.696629	0.629442	0.661333	0.000000	1.080000	3.533333	1.879716	0.490225
6	binary	1020	0.848039	0.635095	0.726280	0.000000	0.973638	4.235501	2.058033	0.552161

Análisis de evaluación de los modelos por clase:

Los resultados muestran que el modelo base (*Experimento 1*) ofrece el mejor desempeño global, especialmente en clases mayoritarias como cobos y búfalos. En cambio, el fine-tuning superficial del *Experimento 2* redujo notablemente la precisión en casi todas las clases, evidenciando que

este enfoque no logra una adaptación efectiva. El *Experimento 3*, al ajustar también capas más profundas (Level4 y Level5), mejoró el rendimiento general, aunque aún con dificultades en clases minoritarias. Finalmente, el *Experimento 4*, que incorporó tasas de aprendizaje diferenciadas y un *scheduler* dinámico, logró el mejor equilibrio entre precisión y generalización, destacando en búfalos, cobos y elefantes. Estos resultados sugieren que la especialización progresiva de capas profundas, más que el reentrenamiento superficial, es clave para mejorar la detección por clase en dominios complejos. Se recomienda como línea futura aplicar técnicas de ajuste focalizado por clase para reforzar el desempeño en especies minoritarias.

Anexo 4: Muestra de inferencia diferenciada por cada modelo generado en las experimentaciones:

Inferencia Fine- tuning: Experimento 2:

Foto general:



Inferencia detección de clase:



Inferencia Fine- tuning: Experimento 3:

Foto general:



Inferencia detección de clase:



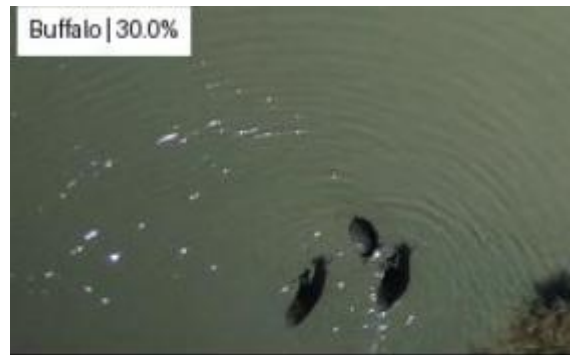
Inferencia Fine- tuning: Experimento 4:

Foto general:



Inferencia detección de clase:





Análisis de inferencias sobre imagen aérea con los modelos generados en las experimentaciones de fine-tuning:

Se realizó un ejercicio comparativo de inferencia sobre una misma imagen seleccionada aleatoriamente utilizando los modelos ajustados en los experimentos 2, 3 y 4. En cuanto a la detección de animales, se evidenció una mejora progresiva en la localización de los individuos, siendo el modelo del *Experimento 4* el que mostró una mayor cobertura y coherencia espacial en la detección de los animales presentes, incluso en condiciones de oclusión parcial.

Por otro lado, los resultados de la clasificación por especie presentaron un comportamiento más variable. El modelo del *Experimento 2* clasificó correctamente la clase "Buffalo" con una alta confianza del 93% mientras que el modelo del *Experimento 3* apenas alcanzó un 20%, y el *Experimento 4* mostró una confianza intermedia del 30%. Esto sugiere que, si bien los ajustes aplicados en el *Experimento 4* beneficiaron la localización, no necesariamente optimizaron el rendimiento del clasificador final en comparación con el modelo del *Experimento 2*.

Los resultados obtenidos evidencian que los ajustes realizados en el modelo durante el *Experimento 4* mejoraron la capacidad de detección de animales en imágenes aéreas, pero no necesariamente incrementaron la precisión en la clasificación por especie la cual fue superior en el modelo del experimento 2. Esto sugiere que existe un equilibrio delicado entre la capacidad de localizar y la de clasificar, y que futuras mejoras deberán considerar estrategias que optimicen ambos componentes de manera conjunta, como por ejemplo, el entrenamiento multi-tarea con pérdida ponderada, en donde se asignan pesos dinámicos a cada tarea según su rendimiento en validación, o preentrenamiento por especie con datasets más balanceados, esto para que el modelo aprenda a discriminar mejor los rasgos visuales entre clases con formas y colores similares.