

Detectção de Uso de Máscara com

KEVIN PONCIANO

MARCELO MARTINS

FELIPPE BASTOS

YOLOv7

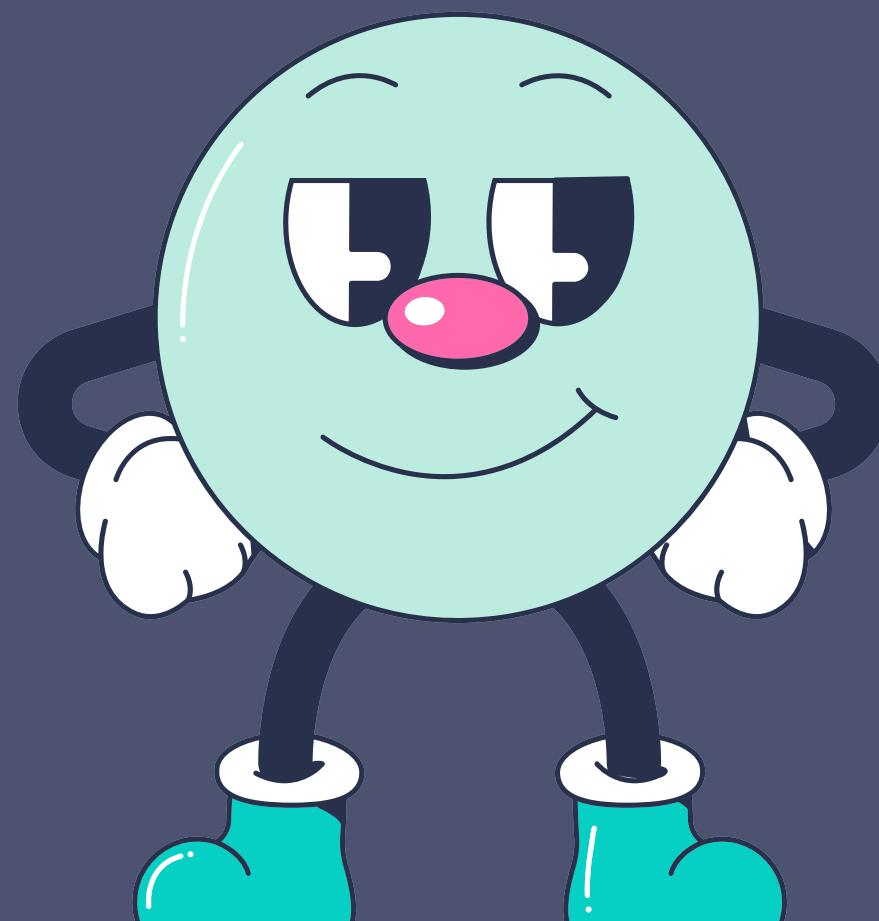
LUCAS GASPAR

MARCELO DE CASTRO

ARTHUR GOMES

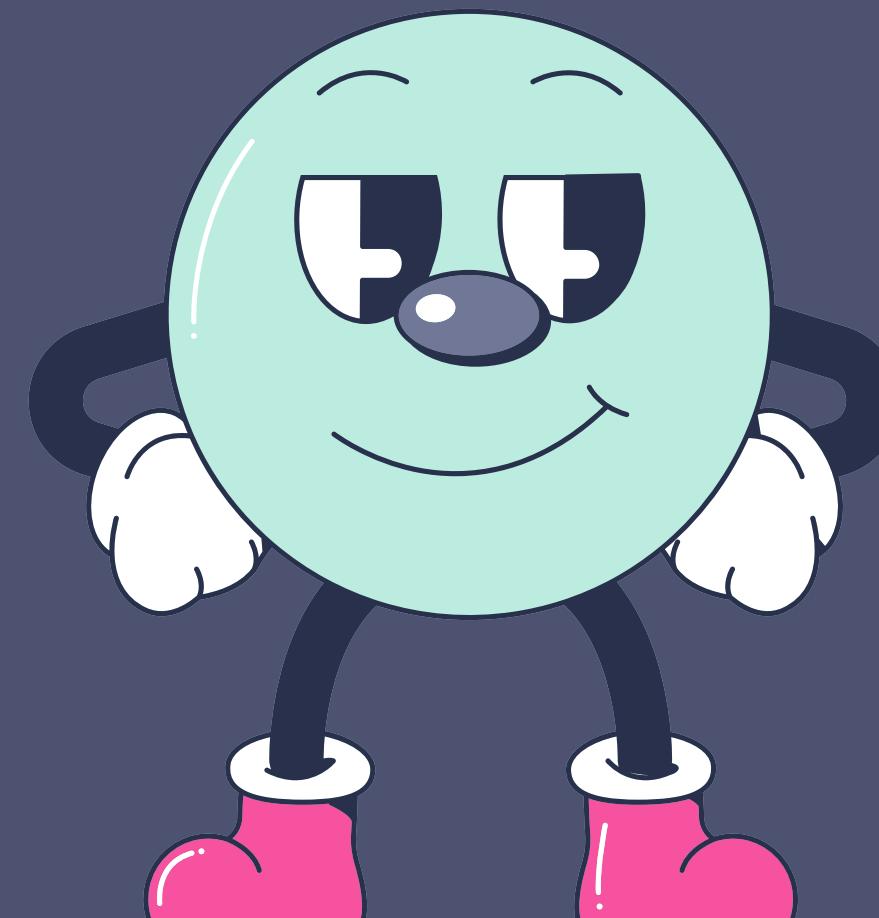
YOLO

Saco de brindes treinável



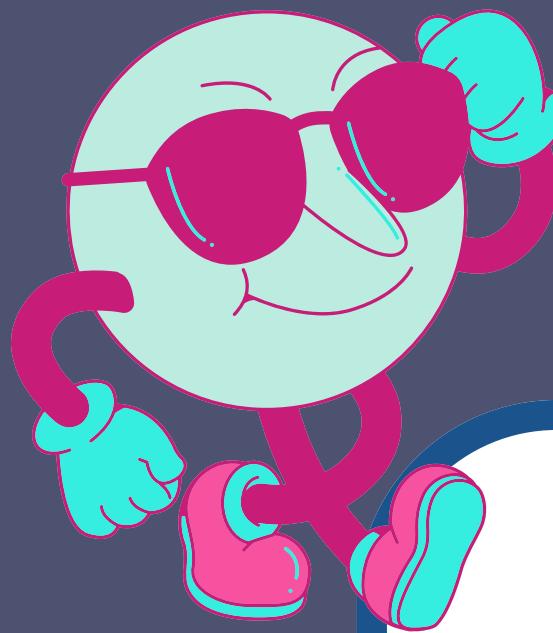
O YOLOv7 é algoritmo de detetor de objetos em tempo real de última geração que ultrapassa todos os detectores de objetos conhecidos, tanto em termos de velocidade como de precisão, na gama de 5 FPS a 160 FPS.

índice

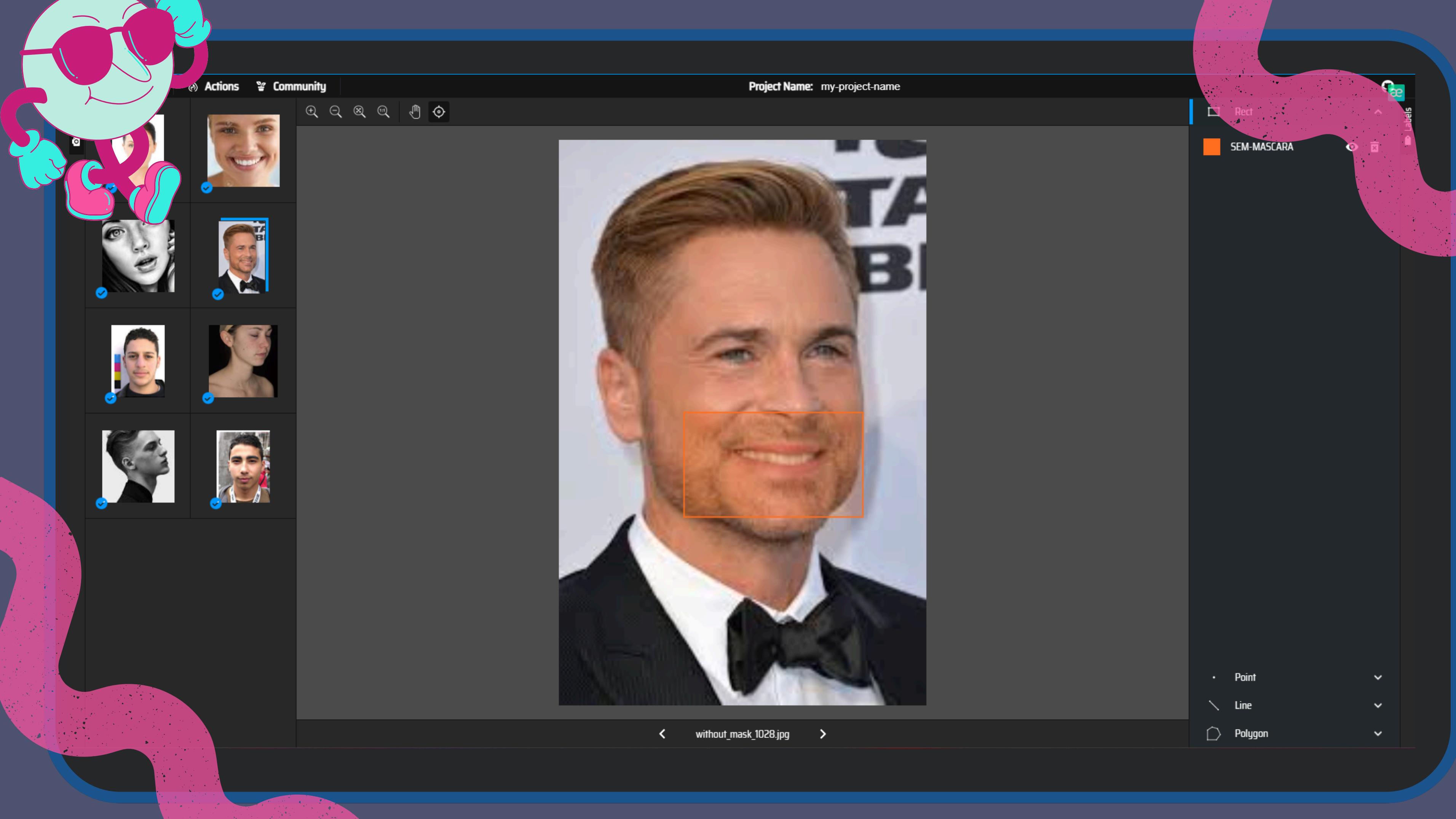


- 1 Preparação dos Dados
- 2 Testes de Sanidade
- 3 Divisão do Dados
- 4 Resultados do Treinamento
- 5 Conclusão





- Utilização do MAKESENSE para a rotulação de 1200 imagens, divididas igualmente entre pessoas com máscara (600) e sem máscara (600).
- Diversidade nas imagens para garantir que o modelo aprenda a reconhecer várias formas, tamanhos e cores de máscaras.



Testes de Sanidade



1



Falha total

2



Duvidoso

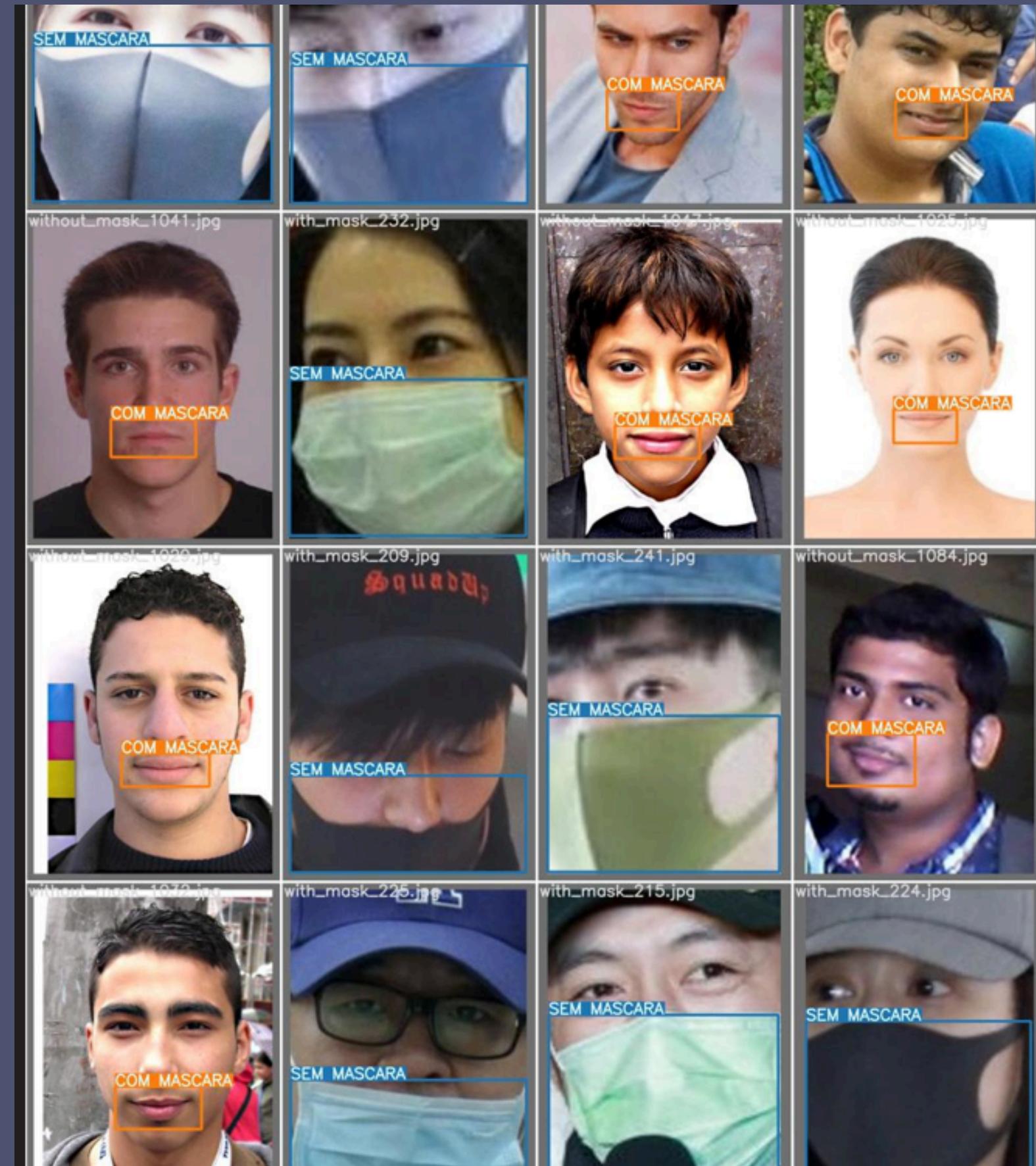
3



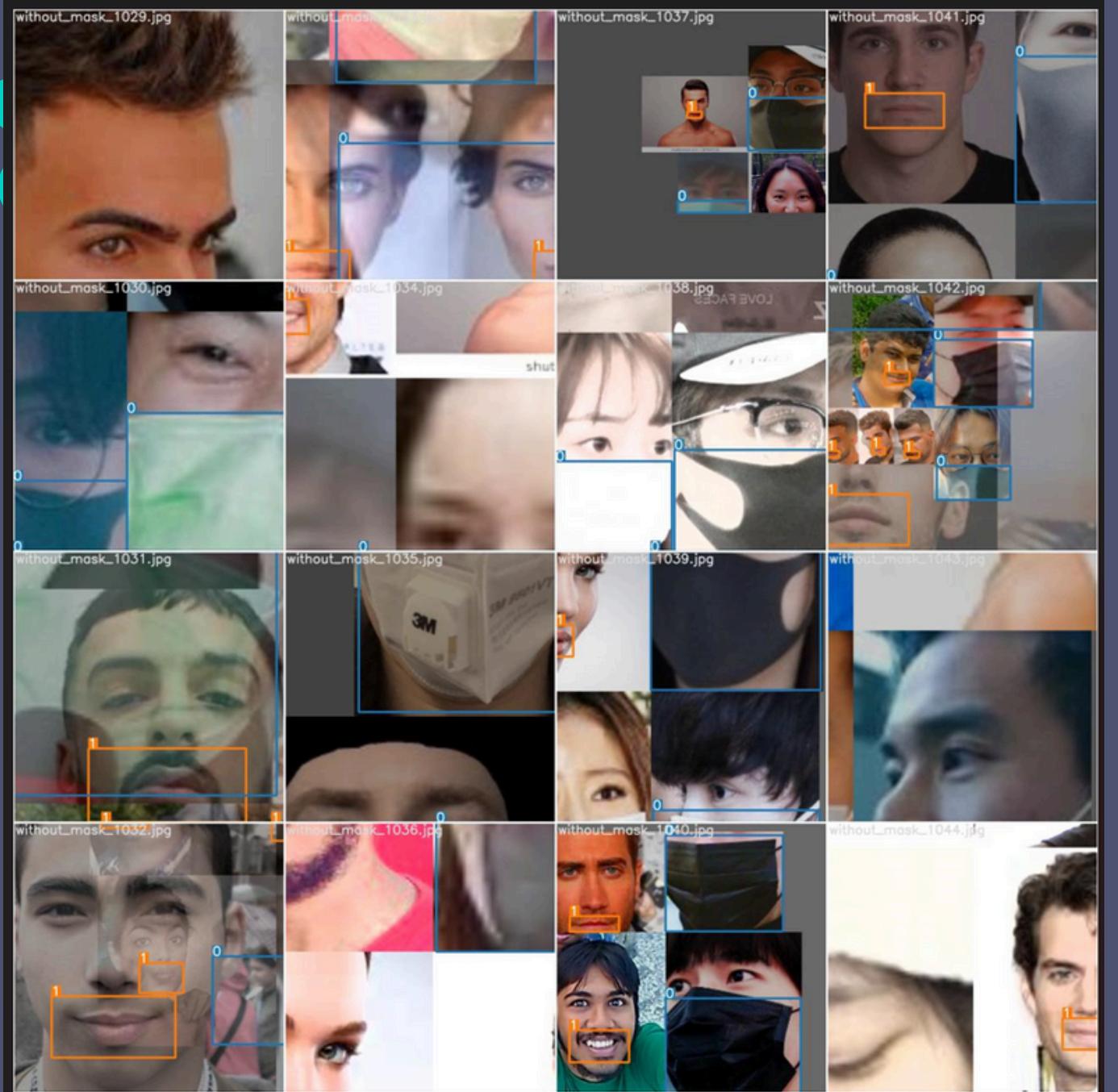
Aceitável



1º Teste Sanidade



2º e 3º Teste Sanidade



2º e 3º Teste Sanidade

Epoch	gpu_mem	box	obj	cls	total	labels	img_size
292/299	6.98G	0.05181	0.000313	0.001243	0.06056	63	640: 100% 4/4 [00:03<00:00, 1.13it/s]
	Class all	Images 120	Labels 127	P 0.822	R 0.726	mAP@.5 0.846	mAP@.5:.95: 100% 2/2 [00:02<00:00, 1.18s/it]
293/299	6.98G	0.05119	0.000344	0.0000928	0.06052	57	640: 100% 4/4 [00:03<00:00, 1.04it/s]
	Class all	Images 120	Labels 127	P 0.89	R 0.667	mAP@.5 0.848	mAP@.5:.95: 100% 2/2 [00:01<00:00, 1.70it/s]
294/299	6.98G	0.0451	0.007694	0.001198	0.05398	50	640: 100% 4/4 [00:02<00:00, 1.37it/s]
	Class all	Images 120	Labels 127	P 0.811	R 0.725	mAP@.5 0.852	mAP@.5:.95: 100% 2/2 [00:01<00:00, 1.69it/s]
295/299	6.98G	0.04881	0.007821	0.001412	0.05805	52	640: 100% 4/4 [00:04<00:00, 1.10s/it]
	Class all	Images 120	Labels 127	P 0.793	R 0.716	mAP@.5 0.836	mAP@.5:.95: 100% 2/2 [00:01<00:00, 1.53it/s]
296/299	6.98G	0.05166	0.00746	0.001588	0.06071	51	640: 100% 4/4 [00:02<00:00, 1.38it/s]
	Class all	Images 120	Labels 127	P 0.766	R 0.709	mAP@.5 0.823	mAP@.5:.95: 100% 2/2 [00:01<00:00, 1.59it/s]
297/299	6.98G	0.05181	0.000327	0.002357	0.0625	75	640: 100% 4/4 [00:03<00:00, 1.33it/s]
	Class all	Images 120	Labels 127	P 0.686	R 0.766	mAP@.5 0.83	mAP@.5:.95: 100% 2/2 [00:01<00:00, 1.21it/s]
298/299	6.98G	0.05432	0.000264	0.001775	0.06436	58	640: 100% 4/4 [00:03<00:00, 1.88it/s]
	Class all	Images 120	Labels 127	P 0.752	R 0.775	mAP@.5 0.849	mAP@.5:.95: 100% 2/2 [00:01<00:00, 1.89it/s]
299/299	6.98G	0.04787	0.000711	0.001566	0.05814	49	640: 100% 4/4 [00:03<00:00, 1.33it/s]
	Class all	Images 120	Labels 127	P 0.747	R 0.749	mAP@.5 0.813	mAP@.5:.95: 100% 2/2 [00:02<00:00, 1.40s/it]
	SEM MASCARA	120	60	0.85	0.662	0.811	0.524
	COM MASCARA	120	67	0.645	0.836	0.815	0.351

300 epochs completed in 0.443 hours.

Optimizer stripped from runs/train/yolov7s/weights/last.pt, 12.3MB
Optimizer stripped from runs/train/yolov7s/weights/best.pt, 12.3MB





Divisão dos Dados

- Dados para Treinamento: 838 imagens foram selecionadas para o treinamento do modelo após exclusão de imagens que poderiam induzir erros.
- Dados para Teste: Separação de 180 imagens para testar a eficácia do modelo em condições similares às reais.
- Dados para Validação: Uso de 180 imagens para validar.

RESULTADOS DO TREINAMENTO

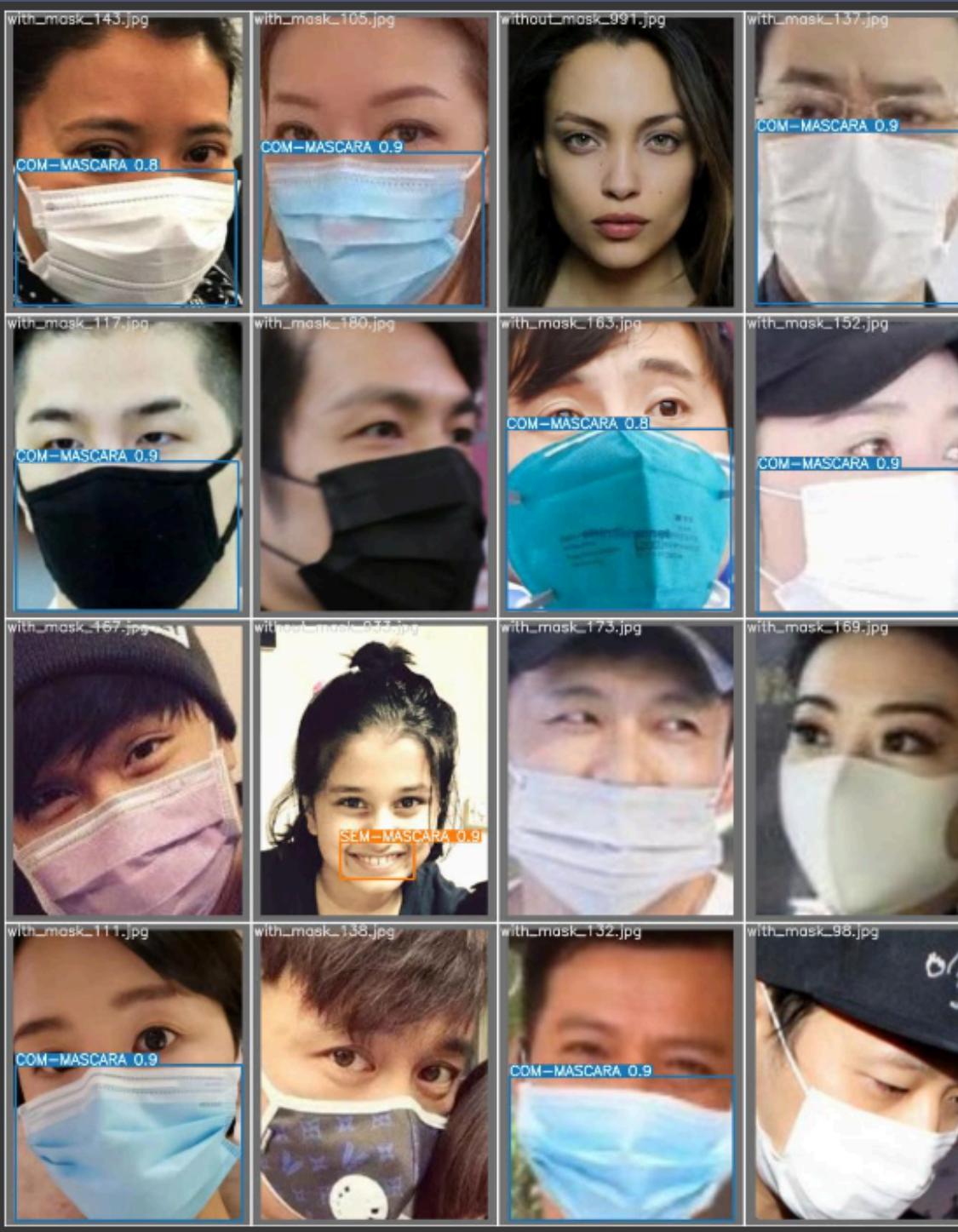
Starting training for 300 epochs...

Epoch	gpu_mem	box	obj	cls	total	labels	img_size
0/299	1.93G	0.06991	0.01664	0.01657	0.1031	4	640: 100% 39/39 [01:06<00:00, 1.70s/it]
	Class	Images	Labels	P		R	mAP@.5 mAP@.5:.95: 100% 5/5 [00:31<00:00, 6.21s/it]
	all	180	187	0.00113	0.454	0.000816	9.9e-05
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
1/299	14.7G	0.06201	0.01268	0.01518	0.08987	6	640: 100% 39/39 [00:45<00:00, 1.17s/it]
	Class	Images	Labels	P		R	mAP@.5 mAP@.5:.95: 100% 5/5 [00:03<00:00, 1.37it/s]
	all	180	187	0.000795	0.457	0.000816	9.9e-05
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
2/299	14.7G	0.06145	0.01114	0.01436	0.08695	176/299	15.3G 0.03455 0.006228 0.0004457 0.04123 6 640: 100% 39/39 [00:46<00:00, 1.20s/it]
	Class	Images	Labels	P		R	mAP@.5 mAP@.5:.95: 100% 5/5 [00:02<00:00, 1.76it/s]
	all	180	187	0.501	0.4	0.882	0.867 0.526
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
3/299	14.7G	0.0603	0.01075	0.0136	0.08465	177/299	15.3G 0.03374 0.005806 0.000371 0.03992 4 640: 100% 39/39 [00:46<00:00, 1.19s/it]
	Class	Images	Labels	P		R	mAP@.5 mAP@.5:.95: 100% 5/5 [00:03<00:00, 1.57it/s]
	all	180	187	0.0235	0.02	0.89	0.894 0.534
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
178/299	15.3G	0.03415	0.006031	0.000639	0.04082	178/299	15.3G 0.03415 0.006031 0.000639 0.04082 5 640: 100% 39/39 [00:47<00:00, 1.22s/it]
	Class	Images	Labels	P		R	mAP@.5 mAP@.5:.95: 100% 5/5 [00:02<00:00, 1.75it/s]
	all	180	187	0.894	0.869	0.874	0.539
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
179/299	15.3G	0.03421	0.006088	0.0005871	0.04089	179/299	15.3G 0.03421 0.006088 0.0005871 0.04089 5 640: 100% 39/39 [00:47<00:00, 1.21s/it]
	Class	Images	Labels	P		R	mAP@.5 mAP@.5:.95: 100% 5/5 [00:02<00:00, 1.76it/s]
	all	180	187	0.938	0.888	0.905	0.538
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
180/299	15.3G	0.0331	0.005628	0.0003054	0.03904	180/299	15.3G 0.0331 0.005628 0.0003054 0.03904 4 640: 100% 39/39 [00:47<00:00, 1.22s/it]
	Class	Images	Labels	P		R	mAP@.5 mAP@.5:.95: 100% 5/5 [00:02<00:00, 1.69it/s]
	all	180	187	0.913	0.908	0.899	0.549
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
181/299	15.3G	0.03392	0.005651	0.001006	0.04058	181/299	15.3G 0.03392 0.005651 0.001006 0.04058 3 640: 100% 39/39 [00:46<00:00, 1.19s/it]
	Class	Images	Labels	P		R	mAP@.5 mAP@.5:.95: 100% 5/5 [00:03<00:00, 1.54it/s]
	all	180	187	0.915	0.901	0.899	0.53
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
182/299	15.3G	0.03352	0.005785	0.0009181	0.04022	182/299	15.3G 0.03352 0.005785 0.0009181 0.04022 3 640: 100% 39/39 [00:47<00:00, 1.22s/it]
	Class	Images	Labels	P		R	mAP@.5 mAP@.5:.95: 100% 5/5 [00:02<00:00, 1.74it/s]
	all	180	187	0.941	0.915	0.923	0.556

TRAIN.PY

RESULTADOS DO TREINAMENTO

TEST.PY



```
# testar
%cd /content/yolov
!bash get_coco.sh
!python test.py --data data/coco.yaml --img 640 --batch 32 --conf 0.8 --iou 0.65 --device 0 --weights best.pt

/content/yolov
Namespace(weights=['best.pt'], data='data/coco.yaml', batch_size=32, img_size=640, conf_thres=0.8, iou_thres=0.65, task=YOLOR 🚀 1fcc842 torch 2.3.0+cu121 CUDA:0 (Tesla T4, 15102.0625MB)

Fusing layers...
IDetect.fuse
/usr/local/lib/python3.10/dist-packages/torch/functional.py:512: UserWarning: torch.meshgrid: in an upcoming release, ...
    return _VF.meshgrid(tensors, **kwargs) # type: ignore[attr-defined]
Model Summary: 208 layers, 6010302 parameters, 0 gradients, 13.0 GFLOPS
Convert model to Traced-model...
traced_script_module saved!
model is traced!

val: Scanning 'data/validacao/validacao.cache' images and labels... 180 found, 0 missing, 0 empty, 0 corrupted: 100% 180
      Class   Images   Labels      P      R      mAP@.5  mAP@.5:.95:  83% 5/6 [00:04<00:00, ...
      return F.conv2d(input, weight, bias, self.stride,
      Class   Images   Labels      P      R      mAP@.5  mAP@.5:.95: 100% 6/6 [00:05<00:00, ...
          all     180     187      1  0.383  0.388  0.304
          COM-MASCARA 180     90      1  0.622  0.627  0.537
          SEM-MASCARA 180     97      1  0.144  0.149  0.0711
Speed: 13.8/4.5/18.3 ms inference/NMS/total per 640x640 image at batch-size 32

Evaluating pycocotools mAP... saving runs/test/exp/best_predictions.json...
loading annotations into memory...
pycocotools unable to run: [Errno 2] No such file or directory: './coco/annotations/instances_val2017.json'
Results saved to runs/test/exp
```

RESULTADOS DO TREINAMENTO



```
# Inferir
%cd /content/yolov
!python detect.py --weights best.pt --conf 0.50 --img-size 640 --source data/inferencia/COM-MASCARA\ \(1\).png
```

```
→ /content/yolov
Namespace(weights=['best.pt'], source='data/inferencia/COM-MASCARA (1).png', img_size=640, conf_thres=0.5, iou_thres=0
YOLOR 🚀 1fcc842 torch 2.3.0+cu121 CUDA:0 (Tesla T4, 15102.0625MB)

Fusing layers...
IDetect.fuse
/usr/local/lib/python3.10/dist-packages/torch/functional.py:512: UserWarning: torch.meshgrid: in an upcoming release,
    return _VF.meshgrid(tensors, **kwargs) # type: ignore[attr-defined]
Model Summary: 208 layers, 6010302 parameters, 0 gradients, 13.0 GFLOPS
Convert model to Traced-model...
traced_script_module saved!
model is traced!

1 COM-MASCARA, Done. (4.6ms) Inference, (616.8ms) NMS
The image with the result is saved in: runs/detect/exp8/COM-MASCARA (1).png
Done. (0.884s)
```

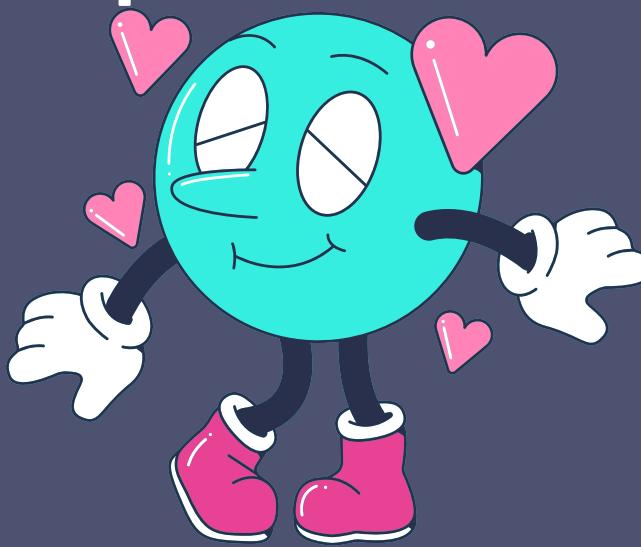
DETECT.PY

CONCLUSÃO



Apesar de todo o esforço dedicado à preparação e rotulagem das 1200 imagens, passando por diversas etapas de ajustes e testes, o desempenho do nosso modelo YOLOv-Tiny na detecção do uso de máscaras não correspondeu às nossas expectativas. Questões como imprecisões nos rótulos das imagens e as limitações do próprio modelo YOLOv-Tiny, que pode não ser tão minucioso quanto outras versões mais complexas, provavelmente impactaram os resultados.

Mesmo enfrentando esses desafios, o projeto nos proporcionou valiosos aprendizados sobre o treinamento de modelos de inteligência artificial para reconhecimento visual. Identificamos como pequenos ajustes podem ter um grande impacto e observamos a importância fundamental de uma preparação minuciosa dos dados antes do início do treinamento. Mesmo que o modelo não tenha apresentado um desempenho perfeito desta vez, as lições aprendidas certamente contribuirão para aprimorarmos em futuros projetos.



Obrigado!

