

Compétition Kaggle

RSNA-MICCAI Brain Tumor Radiogenomic Classification

Agenda

01

Contexte

La compétition et
les données

02

Solution initiale

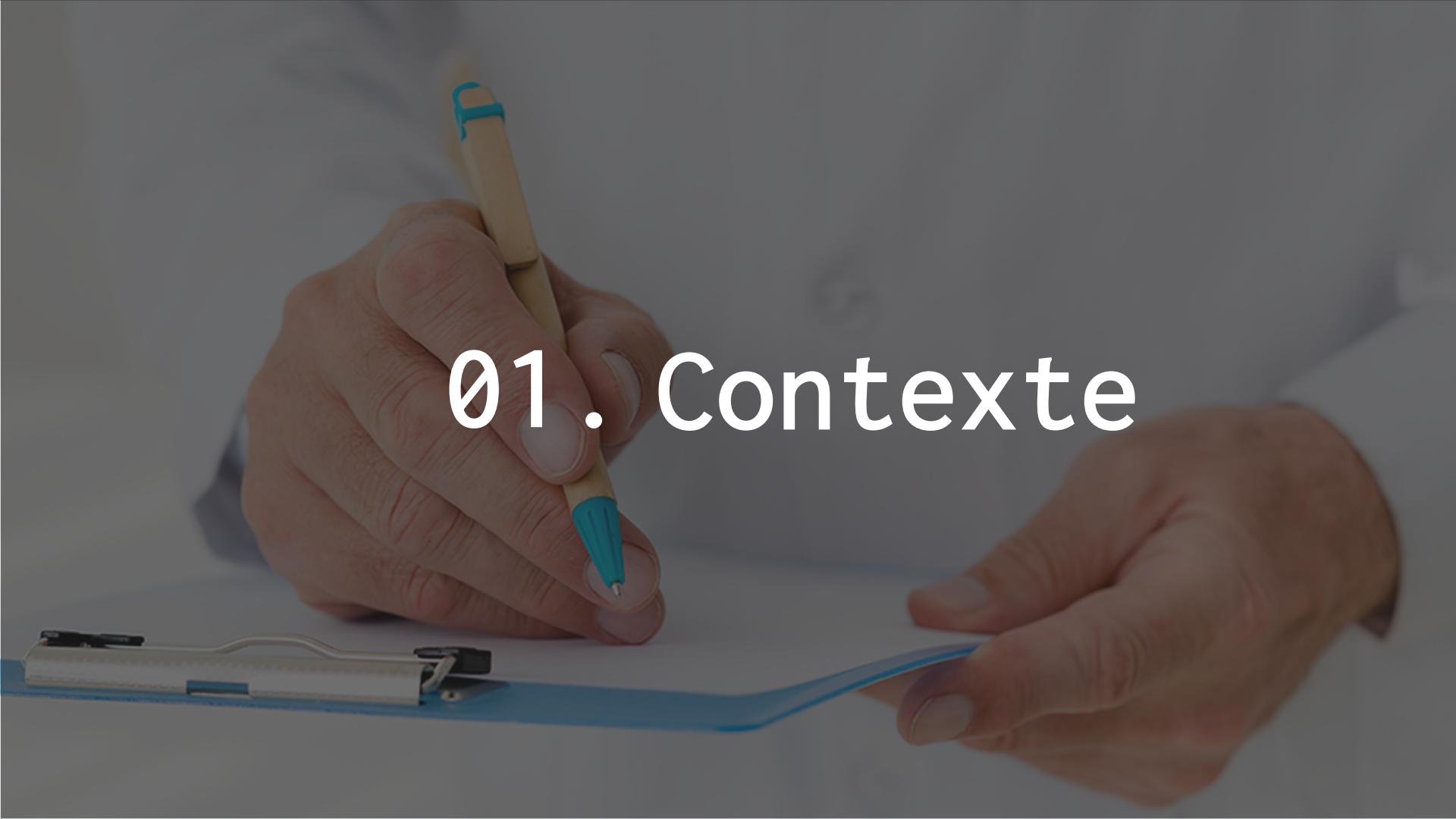
Le modèle et le
kernel initial

03

Implémentation

Les optimisation
mises en œuvre

01. Contexte





Objectif

Prédire la présence du biomarqueur MGMT

Évaluation

AUC

Timing

13 juillet au 25 octobre 2021

Dataset

```
Training/Validation/Testing
|
└── 00000
    |
    └── FLAIR
        |   | Image-1.dcm
        |   | Image-2.dcm
        |   | ...
        |
        └── T1w
            |   | Image-1.dcm
            |   | Image-2.dcm
            |   | ...
            |
            └── T1wCE
                |   | Image-1.dcm
                |   | Image-2.dcm
                |   | ...
                |
                └── T2w
                    |   | Image-1.dcm
                    |   | Image-2.dcm
                    |   | .....
                    |
                    └── 00001
                        |   | ...
```



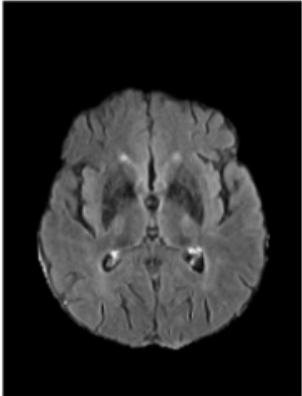
1010 patients



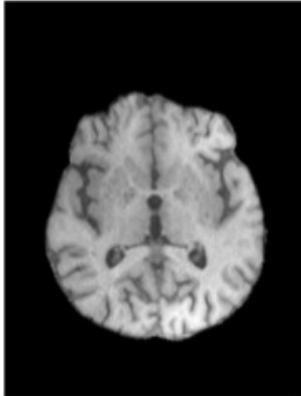
4 types de clichés IRM:
FLAIR, T1w, T1wCE T2

MGMT_value: 0

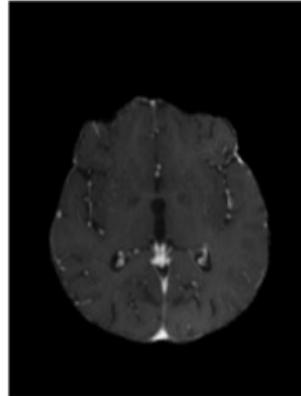
FLAIR



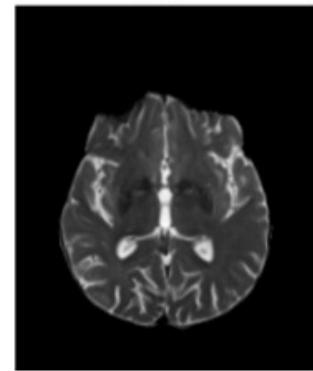
T1w



T1wCE

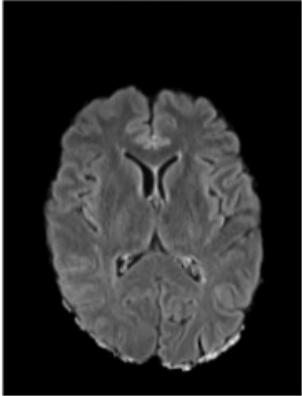


T2w

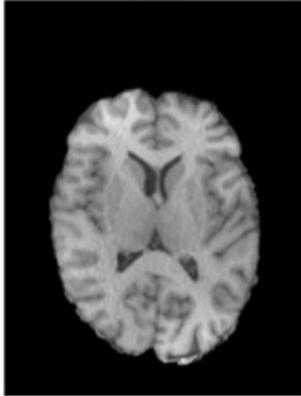


MGMT_value: 1

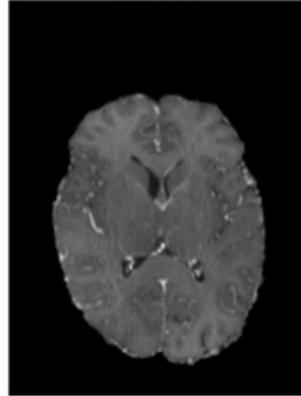
FLAIR



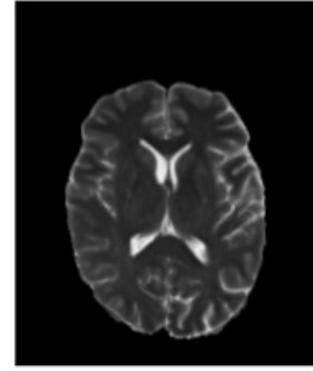
T1w

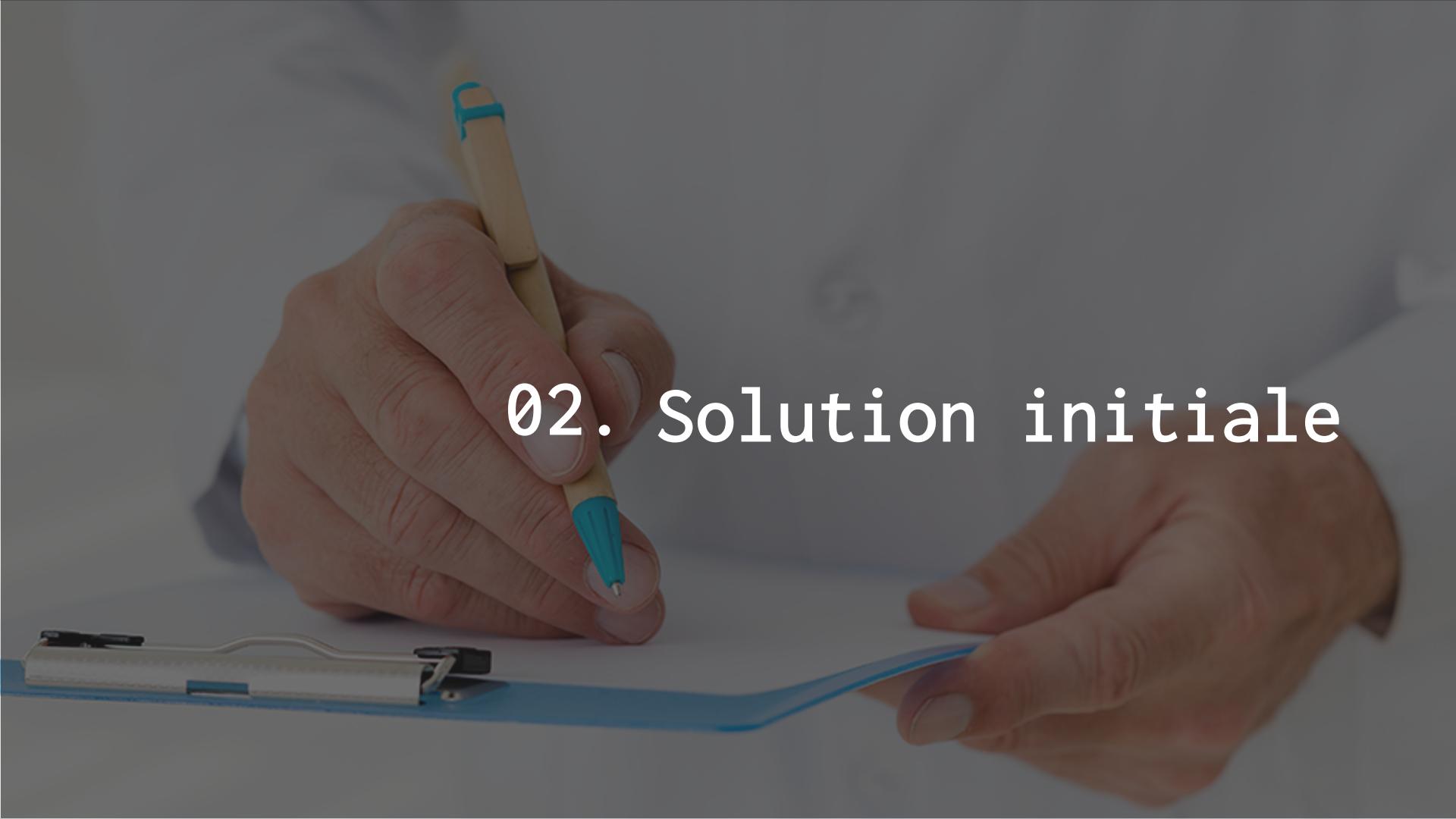


T1wCE



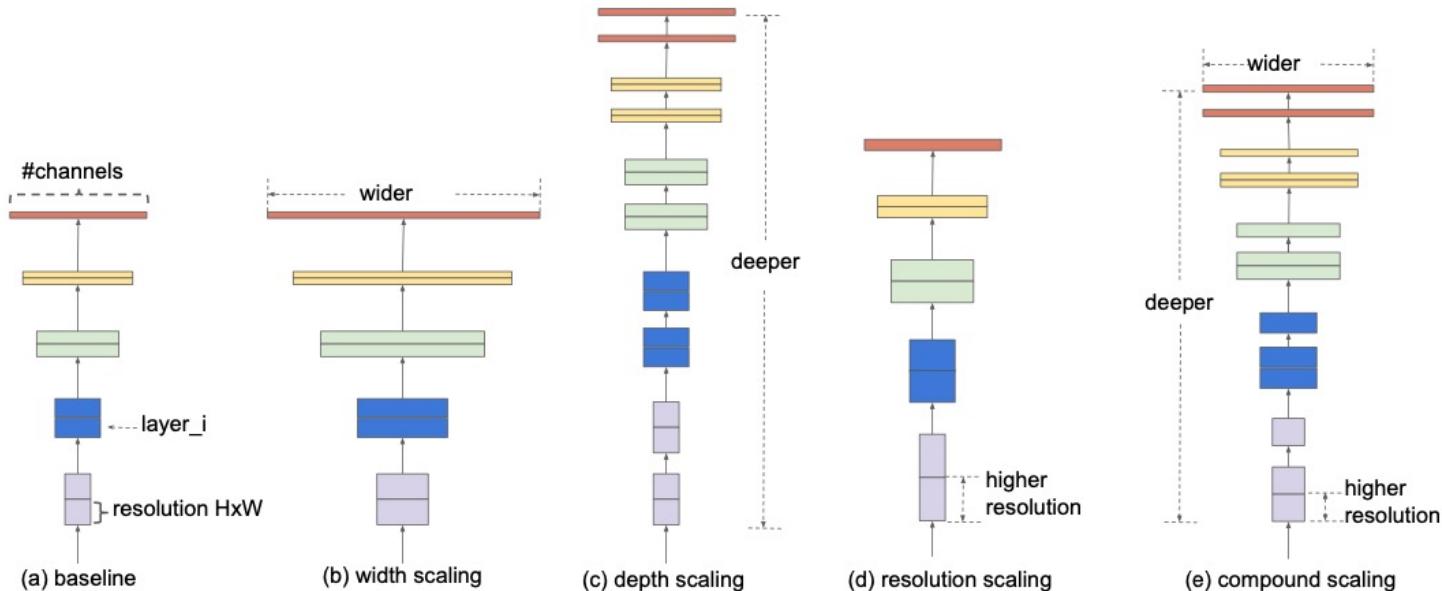
T2w



A close-up photograph of a person's hands. One hand holds a blue pen and is writing on a white sheet of paper held by a blue clipboard. The other hand is visible on the right side of the frame. The background is slightly blurred.

02. Solution initiale

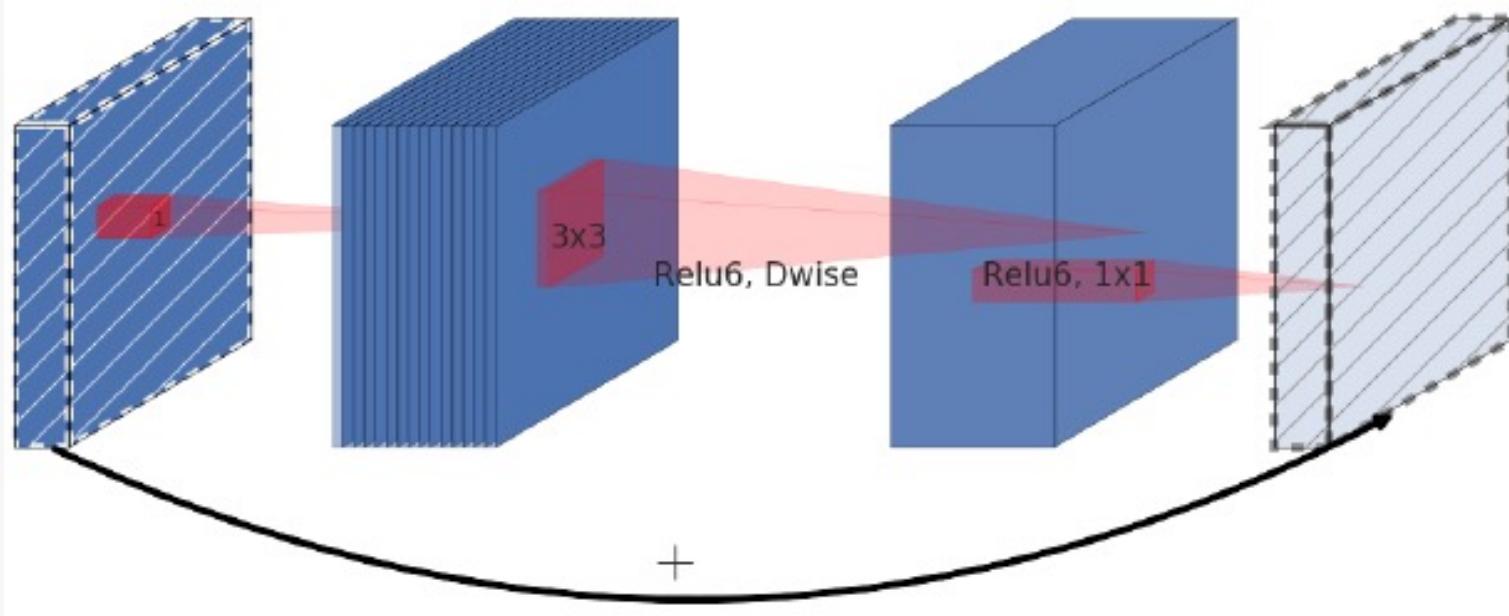
Intuition du modèle EfficientNet



Architecture EfficientNet-B0

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels \hat{C}_i	#Layers \hat{L}_i
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Blocs de convolution MBConv



Modélisation du compound scaling

depth: $d = \alpha^\phi$

width: $w = \beta^\phi$

resolution: $r = \gamma^\phi$

s.t. $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$

$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$

Déclinaisons d'EfficientNet

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPs	Ratio-to-EfficientNet
EfficientNet-B0	77.1%	93.3%	5.3M	1x	0.39B	1x
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	79.1%	94.4%	7.8M	1x	0.70B	1x
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	80.1%	94.9%	9.2M	1x	1.0B	1x
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	81.6%	95.7%	12M	1x	1.8B	1x
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	82.9%	96.4%	19M	1x	4.2B	1x
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	83.6%	96.7%	30M	1x	9.9B	1x
AmoebaNet-C (Cubuk et al., 2019)	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	84.0%	96.8%	43M	1x	19B	1x
EfficientNet-B7	84.3%	97.0%	66M	1x	37B	1x
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

Algorithme de la solution initiale

Calcul_de_prediction() :

Début

 all_test_preds : liste

Pour chaque type de scan :

 train_df ← liens des clichés du type de scan du jeu d'entraînement

 test_df ← liens de clichés du type de scan du jeu de test

 train_g, test_g ← génération_données (train_df, test_df)

 best_model ← entraînement_model(type_cliché, train_g, epochs=5)

 test_pred ← best_model.predict(test_g)

 test_df['pred_y'] ← test_pred

 mean_pred ← Moyenne de test_pred

 test_pred_agg ← **pour chaque patient_id :**

 test_pred_agg ← max(test_df['pred_y'])

Si max(test_df['pred_y']) – mean_pred > mean_pred -

 min(test_df['pred_y'])

Sinon min(test_df['pred_y'])

Fin Si

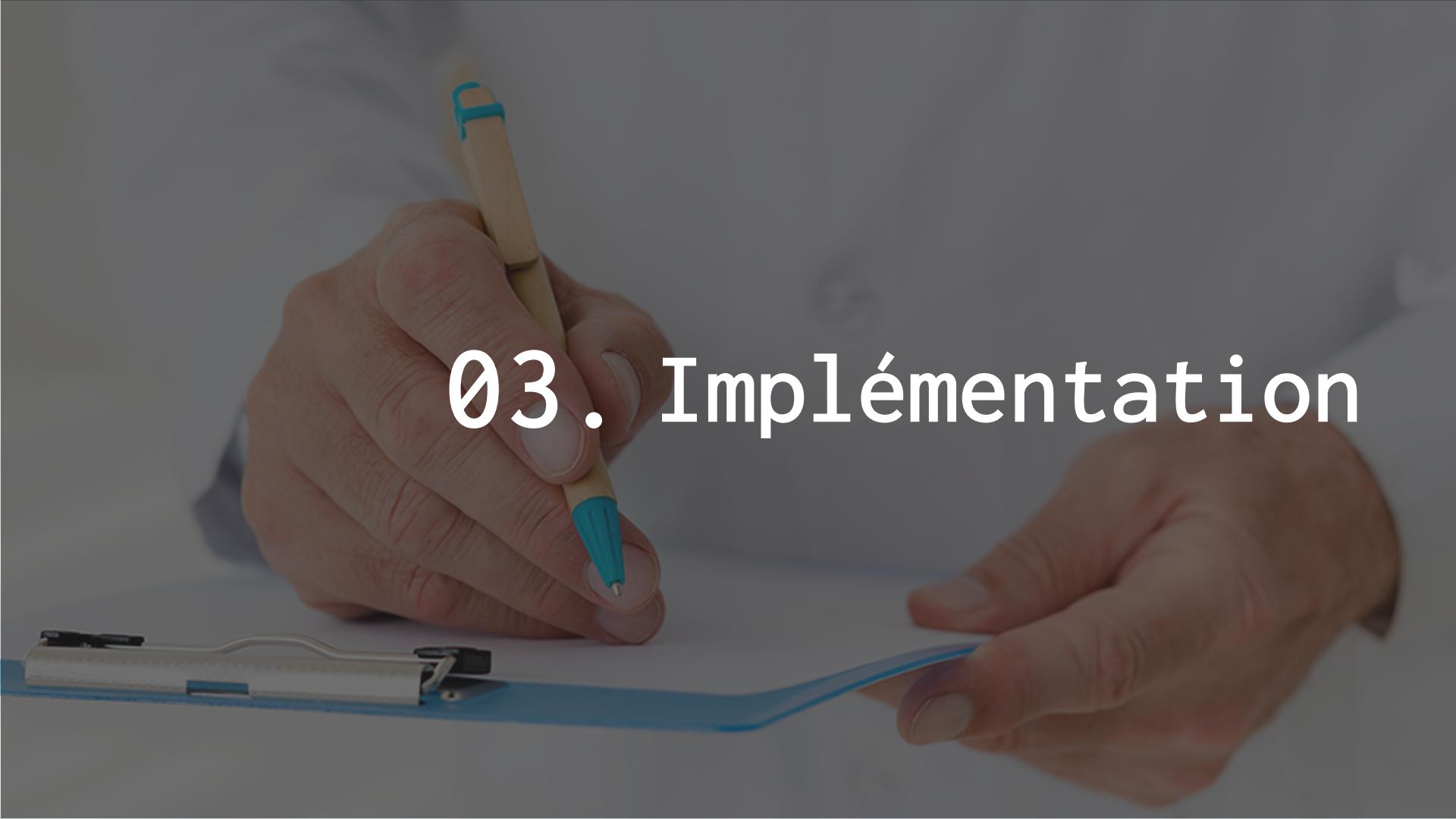
Fin

 all_test_preds ← test_pred_agg

Fin

Fin

03. Implémentation



Optimisation de mise en conformité

Résolution

Passage à 300 x 300 (vs. 224 x 224)

Évaluation des modèles

Partition

90% train 10% validation stratifié sur patient_id et la classe

Appel de la fonction d'évaluation

Lors de l'entraînement

Optimisations des modèles

Allongement de l'entraînement
20 epochs (vs. 5)

Early Stopping
Après 5 epochs sans minimisation de val_loss

Réduction du learning rate
Après 2 epochs sans minimisation de val_loss

Optimisations des modèles

Batch normalization

Avant les deux couches fully connected (vs. Première couche)

Drop out

Avant les deux couches fully connected et passage à 40% (vs. 10%)

Test de Batch sizes

32, 64, 128 (vs. 512)

Comparaison des batch sizes

Batch size	AUC Publique
32	0,61945
64	0,57875
128	0,65909

Résultats solutions initiale vs. optimisée

	Batch size	AUC publique	Classement provisoire	AUC privée	Classement final
Initiale	512	0,62103	906	0,55150	166
Optimisée	128	0,65909	679	0,61732	3

Merci

Des questions ?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik** and illustrations by **Stories**

cedricsoares@me.com
+33 6 09 25 47 45