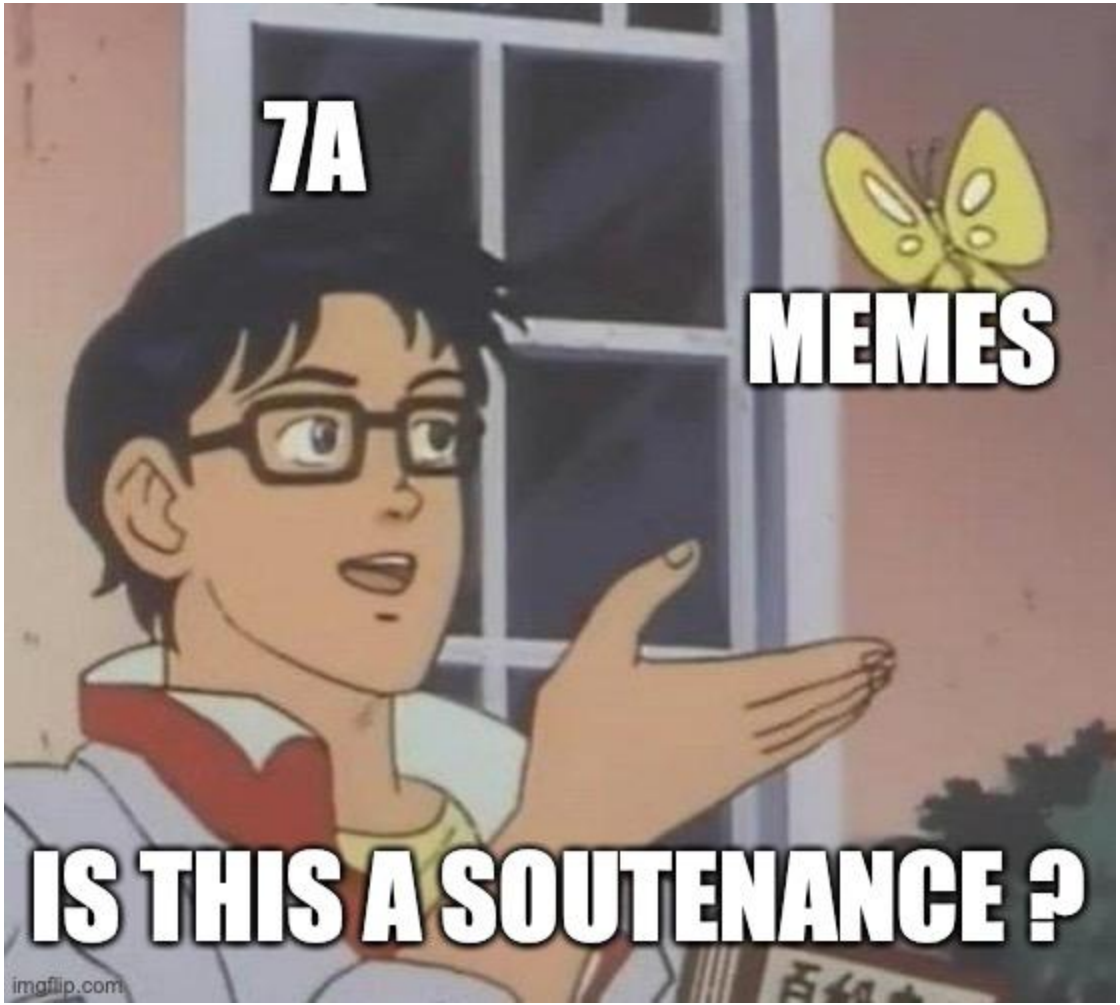
Projet Crowd Counting Code Framework

Groupe 7A
Chi Huynh, Tu Duyen Nguyen, Paul-Henri Pinart


7A
MEMES
IS THIS A SOUTENANCE ?
imgflip.com

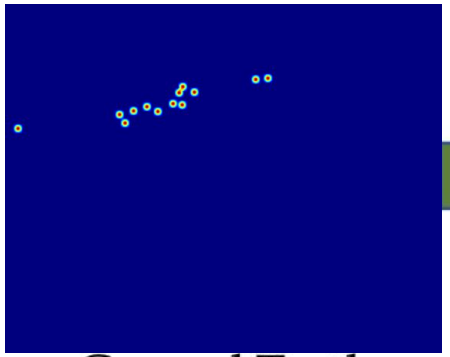
C^3 Crowd Counting Code Framework

- Simplification du process dataloading, train, test, loss
- Variété et hétérogénéité de datasets
- Repo GitHub relativement ancien (last commit il y 5a)

Notre TO-DO List :

- Nettoyer un peu le code pour le rendre utilisable avec Colab
- Tester avec quelques datasets
- Implémenter la gaussian/laplacian loss
- Implémenter la bayesian loss

Input Image



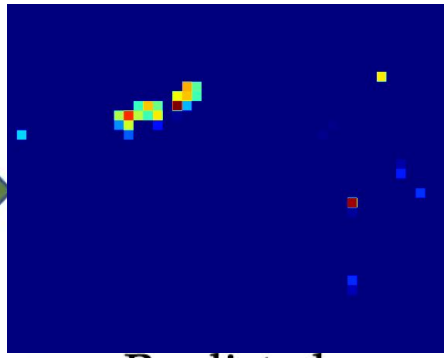
Ground Truth

Image Preprocessing

Ground Truth Preprocessing

Neural Network Model

Loss Function



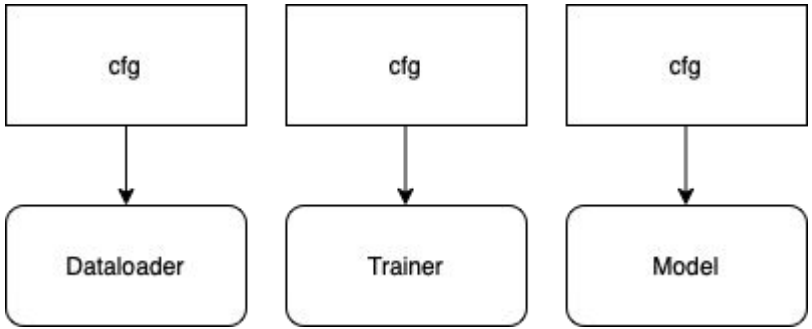
Predicted Density Map

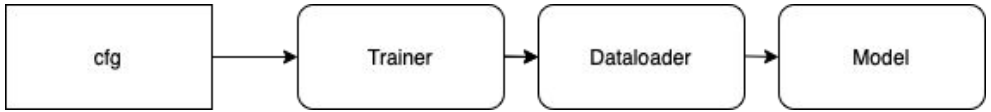
$\mathcal{L}$

Loss Value

Comment l'a-t-on rendu "compatible" avec Colab ?

- Reformatage (plus agréable à lire)
- Refactoring: passer les paramètres depuis un seul point





Premier pas: tester le test()

Compléter les résultats manquants sur WE et UCF-CC-50 (tout en test)

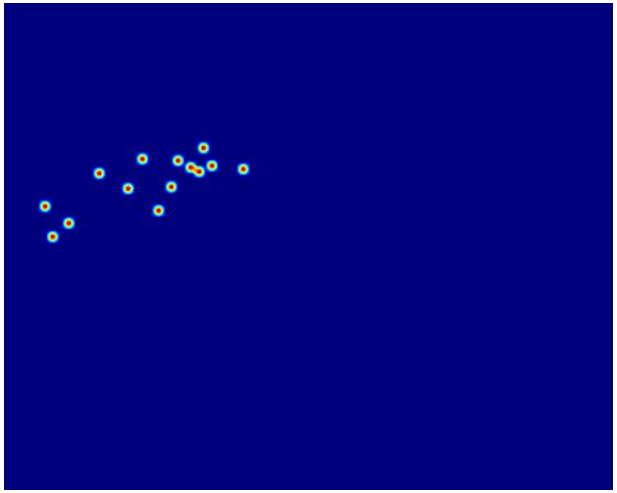
Problème: le train est très (trop) lent

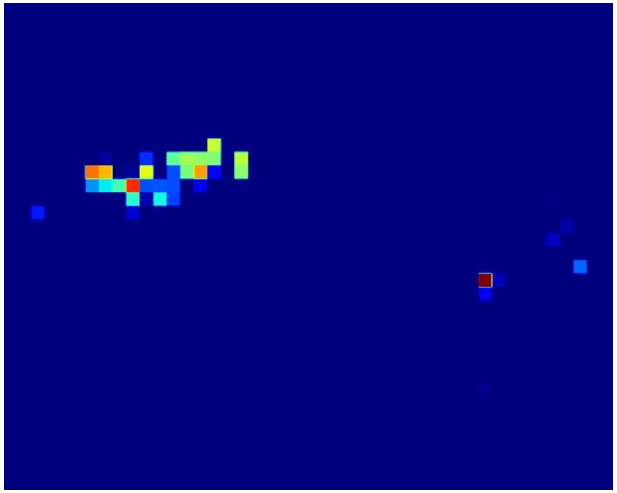
Idée: modèles préentraînés sur GCC

Problème bis: les datasets sont différents

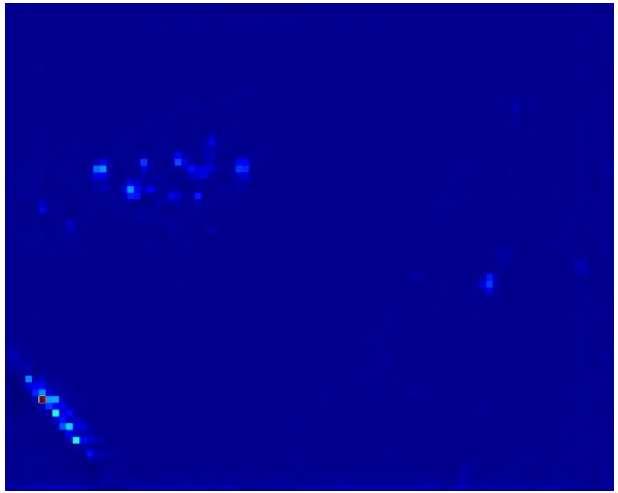
Interprétation: pas de fine tuning, particularités des datasets

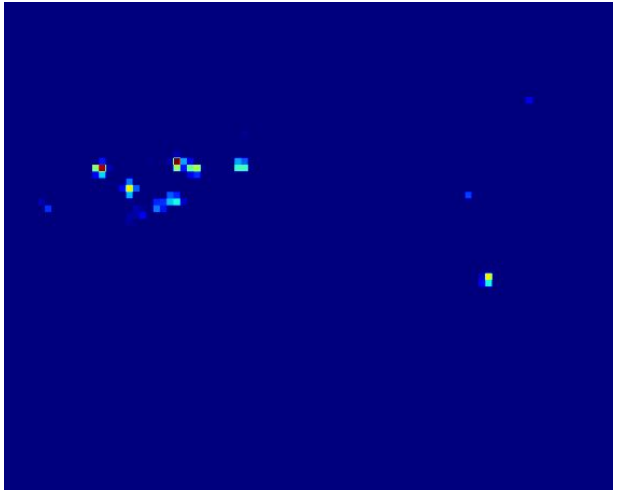
	UCF CC 50	WE s1	WE s2
CSRNet	1271.10 / 2498704	8.27 / 114.10	34.24 / 1468.99
ResNet50	1230.65 / 2405578	10.09 / 201.88	44.96 / 2353.03
ResNet101	**1042.20 / 1831821**	5.83 / 83.36	25.29 / 1004.66
AlexNet	1097.51 / 2005878	**5.57 / 70.97**	**16.31 / 406.67**

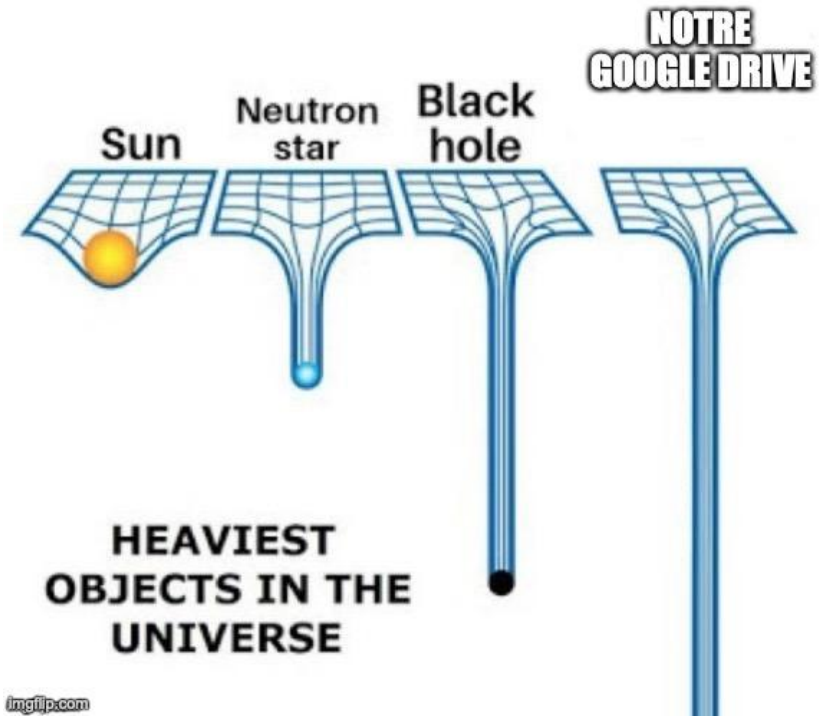


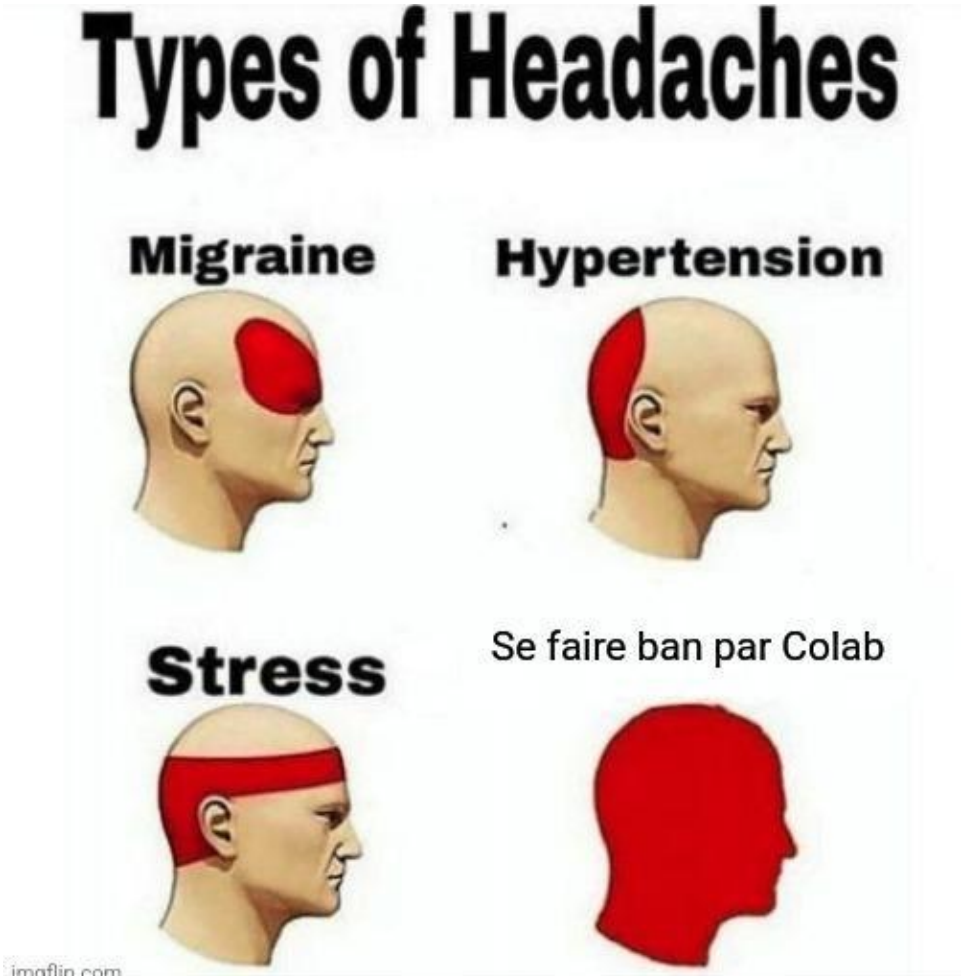


GT: 13
AlexNet: 9.11
Res50: 4.56
CSRNet: 15.20






NOTRE GOOGLE DRIVE
Sun
Neutron star
Black hole
HEAVIEST OBJECTS IN THE UNIVERSE
imgflip.com

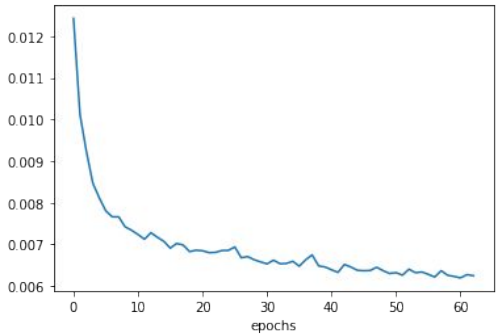

Types of Headaches
Migraine
Hypertension
Stress
Se faire ban par Colab
imgflip.com

Aleatoric Loss: Gaussian et Laplacian Loss

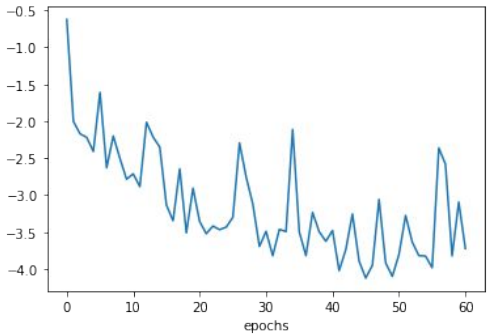
But : modéliser l'incertitude épistémique (via le dropout) et aléatoire (via la prédiction de la variance).

$$\text{lloss} = \frac{1}{2}\left(\log\left(\text{var}\right) + \frac{|\text{input - target}|}{\text{var}}\right) \qquad \text{gloss} = \frac{1}{2}\left(\log\left(\text{var}\right) + \frac{(\text{input - target})^2}{\text{var}}\right)$$

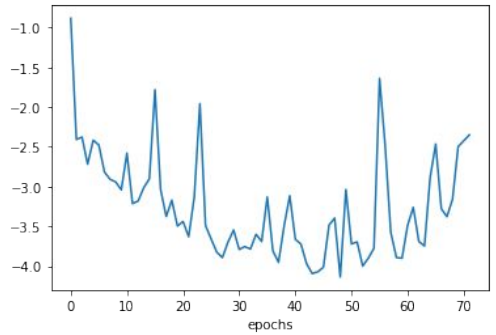
Modification du modèle et du trainer pour prédire la variance (en fait la log-variance). Ici : CSRNet.



MSE



Laplacian (lr = 1e-5)



Gaussian (lr = 1e-5)

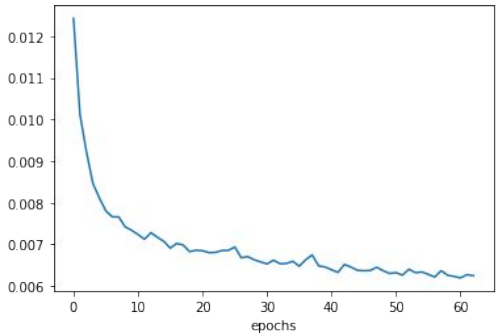
Problèmes de stabilité numérique : utilisation d'un epsilon, modification du learning rate ?

Aleatoric Loss: Gaussian et Laplacian Loss

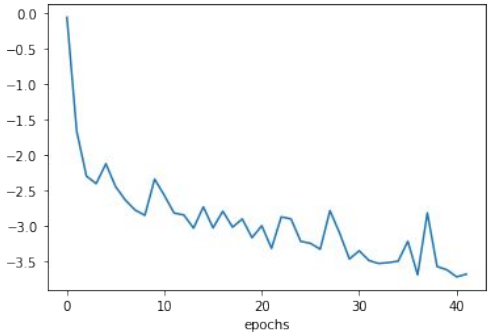
But : modéliser l'incertitude épistémique (via le dropout) et aléatoire (via la prédiction de la variance).

$$\text{lloss} = \frac{1}{2}\left(\log\left(\max\left(\text{var},\ \text{eps}\right)\right) + \frac{|\text{input - target}|}{\max\left(\text{var},\ \text{eps}\right)}\right) \qquad \text{gloss} = \frac{1}{2}\left(\log\left(\max\left(\text{var},\ \text{eps}\right)\right) + \frac{(\text{input} - \text{target})^2}{\max\left(\text{var},\ \text{eps}\right)}\right)$$

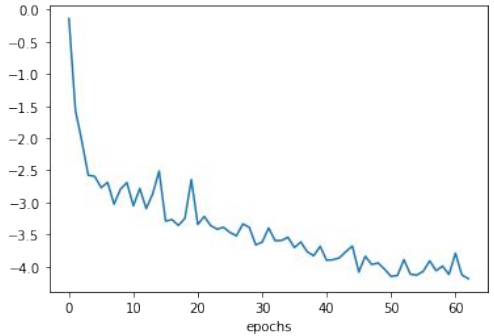
Modification du modèle et du trainer pour prédire la variance (en fait la log-variance). Ici : CSRNet.



MSE



Laplacian (lr = 5e-6)

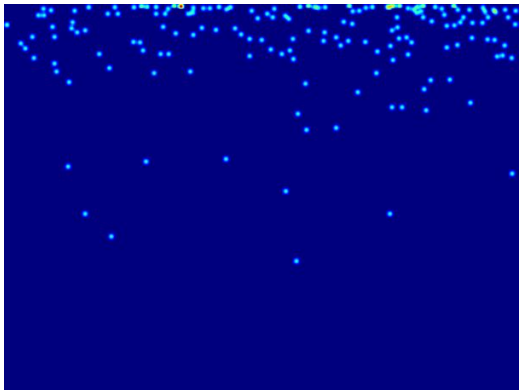


Gaussian (lr = 5e-6)

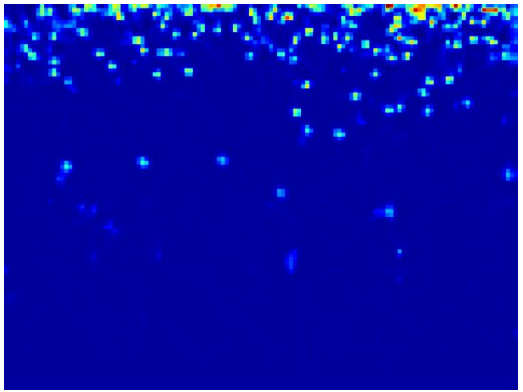
Problèmes d'instabilité numérique : **utilisation d'un epsilon, modification du learning rate**

Aleatoric Loss: résultats

MSE loss : bonnes prédictions, le modèle localise bien les personnes



Ground truth : 164



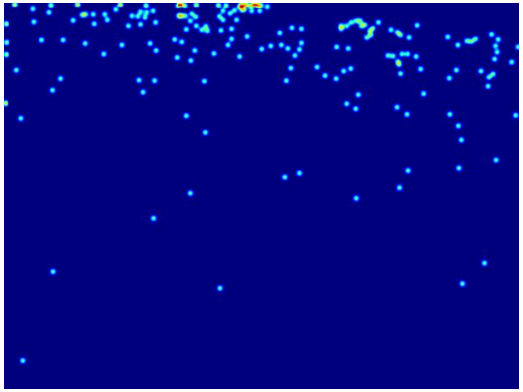
Prédiction : 158

Image

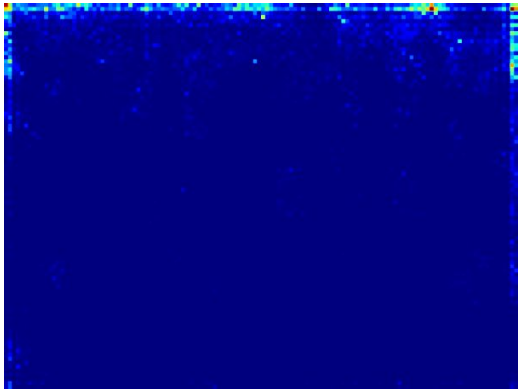
Mean average error : 18.05 ; Mean squared error : 28.75

Aleatoric Loss: résultats

Laplacian loss : prédictions sous-évaluées systématiquement et mauvaise localisation



Ground truth : 180



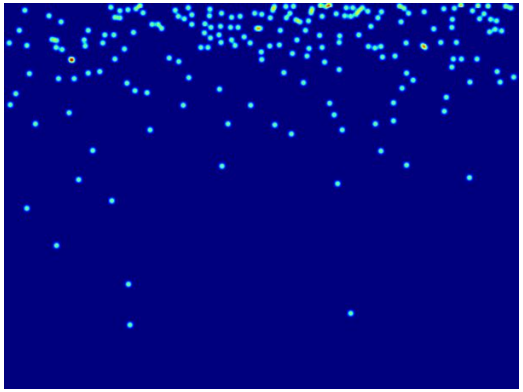
Prédiction : 31

Image

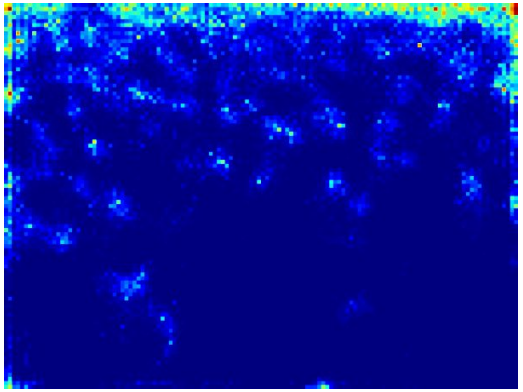
Mean average error : 95.80 ; Mean squared error : 130.16

Aleatoric Loss: résultats

Gaussian loss : les prédictions sont meilleures



Ground truth : 195



Prédiction : 158

Image

Mean average error : 46.15 ; Mean squared error : 63.59

Bayesian Loss

Construction de fct de vraisemblance : $p(\mathbf{x} = \mathbf{x}_m | y = y_n) = \mathcal{N}(\mathbf{x}_m; \mathbf{z}_n, \sigma^2 \mathbf{1}_{2\times 2})$

$$p(y_n|\mathbf{x}_m) = \frac{p(\mathbf{x}_m|y_n)p(y_n)}{p(\mathbf{x}_m)} = \frac{p(\mathbf{x}_m|y_n)p(y_n)}{\sum_{n=1}^{N} p(\mathbf{x}_m|y_n)p(y_n)}$$

$$= \frac{p(\mathbf{x}_m|y_n)}{\sum_{n=1}^{N} p(\mathbf{x}_m|y_n)} = \frac{\mathcal{N}(\mathbf{x}_m; \mathbf{z}_n, \sigma^2 \mathbf{1}_{2\times 2})}{\sum_{n=1}^{N} \mathcal{N}(\mathbf{x}_m; \mathbf{z}_n, \sigma^2 \mathbf{1}_{2\times 2})}$$

$$E[c_n] = \sum_{m=1}^{M} p(y_n|\mathbf{x}_m)\mathbf{D}^{est}(\mathbf{x}_m)$$

$$\mathcal{L}^{Bayes} = \sum_{n=1}^{N} \mathcal{F}(1 - E[c_n])$$

Bayesian+ Loss

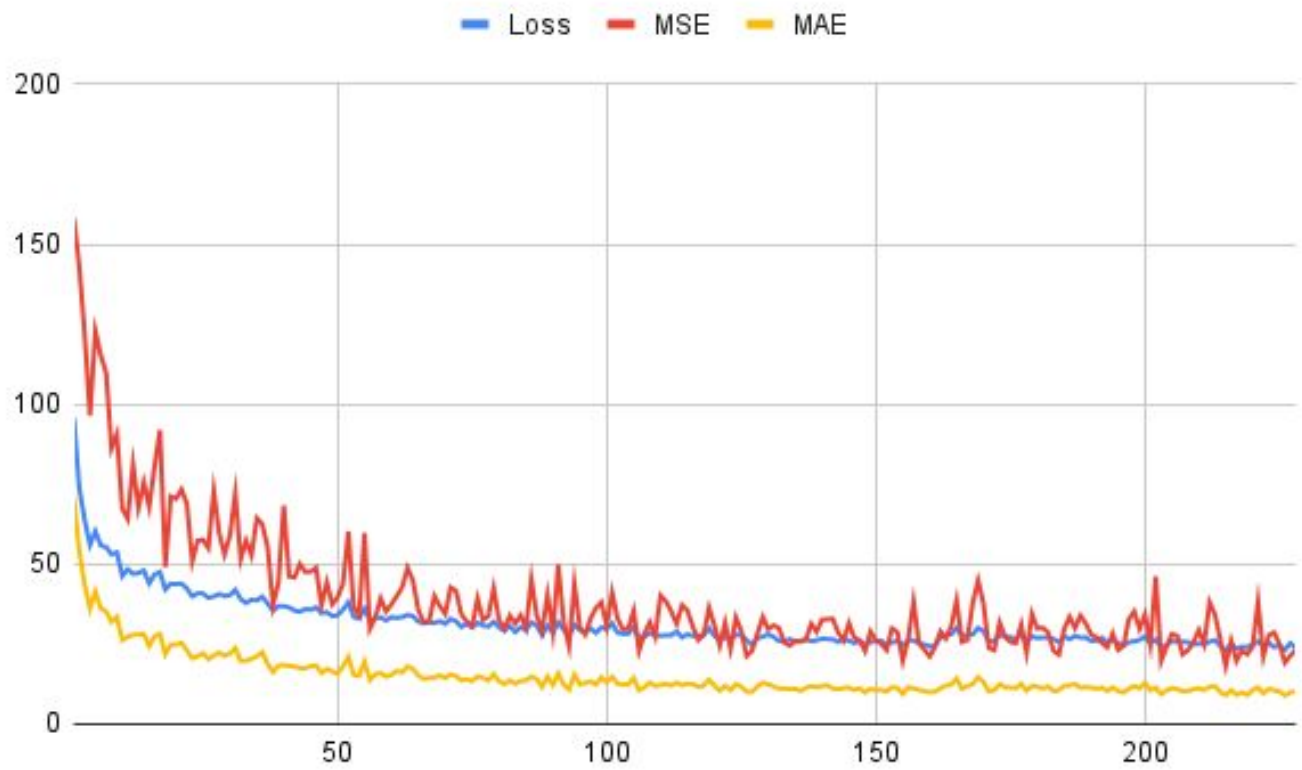
Améliore la précision des prédictions de la densité de foules en prenant en compte le contexte global de l'image.

$$E[c_n] = \sum_{m=1}^{M} p(y_n|\mathbf{x}_m)\mathbf{D}^{est}(\mathbf{x}_m)$$

$$E[c_0] = \sum_{m=1}^{M} p(y_0|\mathbf{x}_m)\mathbf{D}^{est}(\mathbf{x}_m)$$

$$\mathcal{L}^{Bayes+} = \sum_{n=1}^{N} \mathcal{F}(1 - E[c_n]) + \mathcal{F}(0 - E[c_0])$$

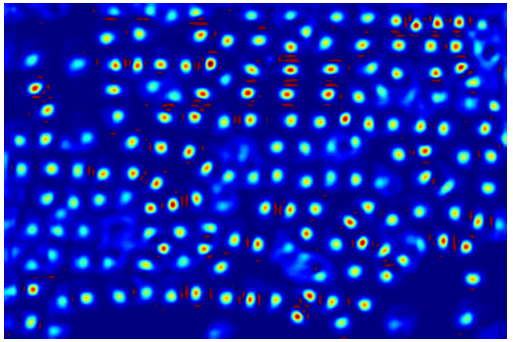
Résultats

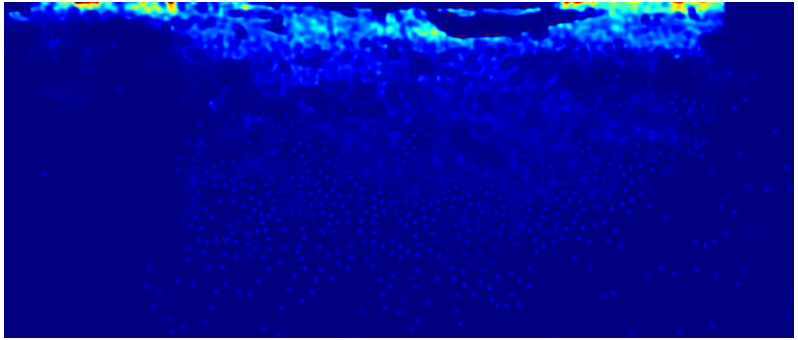


Résultats

Ground truth : 191

Prediction : 179.5740





Prediction : 4627.364

Ground truth : 4535


Dépasser la limite GPU sur Colab
Créer un 2e compte pour contourner le ban
Se refaire ban
Se refaire ban
imgflip.com


VOYONS COMMENT VA LA LOSS APRÈS 10H DE TRAIN
BAN DE COLAB
imgflip.com

Conclusion

TODO en suspens: semi-supervised learning pour intégrer FUDAN-UCC

Importance du datacleaning

Limitations matérielles et techniques

Transfer Learning ?

Références

Aleatoric loss: **Kendall and Gal, 2017**

Bayesian loss: **Ma, Wei et al., 2019**

Utiliser un notebook Colab pour profiter des GPU

Se faire ban par Colab parce que le train est trop gourmand

Créer un 2e compte Google pour contourner le ban

Se faire ban le 2e compte

UCF50_results_CSRNet
MAE: 1271.108480355835
MSE: 2498704.743848537
AVG gt: 1278.48
AVG pred: 7.371519644165039

UCF50_results_VGG
MAE: 1223.2695530925753
MSE: 2289639.0810503196
AVG gt: 1278.48
AVG pred: 55.210446907424924

WE_202201_results_Res101
MAE: 25.29493501790364
MSE: 1004.663851186799
AVG gt: 89.16666666666667
AVG pred: 107.17970406087238

WE_202201_results_AlexNet
MAE: 16.31345788574219
MSE: 406.67130495879513
AVG gt: 89.16666666666667
AVG pred: 101.27052526855468

UCF50_results_Res101
MAE: 1042.2021305342673
MSE: 1831821.3642919997
AVG gt: 1278.48
AVG pred: 236.27786946573258

UCF50_results_Res101_SFCN
MAE: 1259.4200980109572
MSE: 2469608.22421882
AVG gt: 1278.48
AVG pred: 19.05990198904276

WE_104207_results_Res101
MAE: 5.834239532647012
MSE: 83.36656273465799
AVG gt: 15.184873949579831
AVG pred: 9.913246650936221

WE_104207_results_AlexNet
MAE: 5.573125330820804
MSE: 70.96603933029373
AVG gt: 15.184873949579831
AVG pred: 10.644829781155625

UCF50_results_Res50
MAE: 1230.651065873289
MSE: 2405578.115194637
AVG gt: 1278.48
AVG pred: 47.828934126710884

WE_104207_results_Res101_SFCN
MAE: 6.997081488601299
MSE: 113.2558052833404
AVG gt: 15.184873949579831
AVG pred: 8.661885402262714

WE_104207_results_CSRNet
MAE: 8.2723512383469
MSE: 114.10240397901082
AVG gt: 15.184873949579831
AVG pred: 16.12645156988577

WE_202201_results_Res50
MAE: 44.95641501871744
MSE: 2353.0346968716603
AVG gt: 89.16666666666667
AVG pred: 44.21025164794922

UCF50_results_AlexNet
MAE: 1097.5111670623778
MSE: 2005878.3401329073
AVG gt: 1278.48
AVG pred: 188.12289387512206

WE_202201_results_Res101_SFCN
MAE: 25.77570037841797
MSE: 1074.2931187904117
AVG gt: 89.16666666666667
AVG pred: 86.36351153564452

WE_202201_results_CSRNet
MAE: 34.24720908610026
MSE: 1468.9903898972566
AVG gt: 89.16666666666667
AVG pred: 54.9194575805664

WE_104207_results_Res50
MAE: 10.087355123167278
MSE: 201.88310431312829
AVG gt: 15.184873949579831
AVG pred: 5.432457569907694