Google Landmark Recognition Challenge

Lupașcu Marian **Coordonator: Iulia Duță**

Introducere

In primăvara lui 2018 Google a lansat Landmark Recognition Challenge, o competiție ce presupune antrenarea unui **clasificator pentru recunoașterea clădirilor si monumentelor celebre de pe mapamond**. In acest proiect am implementat o serie de modele printre care: **VGG-16**[1] preantrenat pe ImageNet la care am antrenat doar clasificatorul de la final si **VGG-16** preantrenat modificat la care am adăugat un **modul de atenție**.

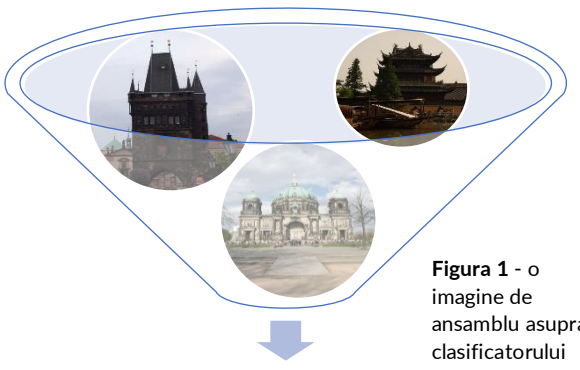

Figura 1 - o imagine de ansamblu asupra clasificatorului

Arhitectura rețelei

Pe date s-a facut **augmentare**, anume: pe train: rotație la $\pm 45°$, random crop la 224×224, schimbare de iluminare de pana la 25% si random flip, pe validare si test: center crop la 224×224 si rotație la $\pm 45°$. Arhitectura folosita este VGG-16 la care s-a modificat secvența de clasificare de la final, folosind **Dropout** după fiecare layer de fully-connected pentru **regularizare**. Antrenarea modelului se face doar pe secvența de **clasificare**, secvența de **feature extraction** rămânând înghețata pe parcursul antrenării, deoarece VGG-16 fiind deja preantrenat pe 1000 de clase aste de așteptat ca aceasta parte sa returneze o serie de feature-uri bine definite, urmând sa fie clasificate.

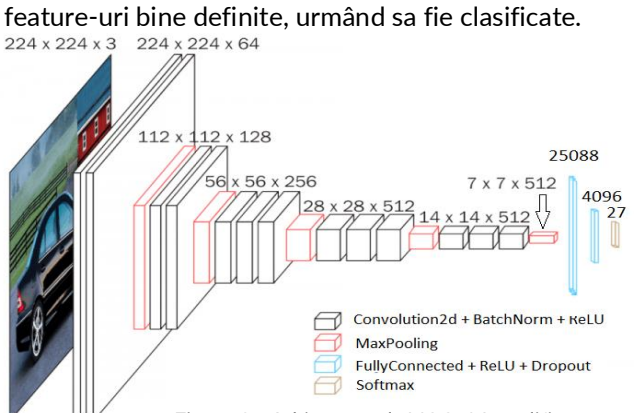

Figura 3 - Arhitectura de VGG-16 modificata

Descrierea Datasetului

Datasetul este format din aproximativ 12 milioane de imagini cu 14952 clădiri (clase), pe care le-am descărcat de internet apoi am luat cele mai definite 27 de clase, adică aproximativ 11.5% din dimensiunea datasetului (46GB) si 0.18% din numărul de clase. O histograma care prezinta greutatea celor mai definite 100 de clase din dataset este descrisa. Se poate observa cu ușurința ca datele sunt debalansate variind de la 49091 exemple pentru clasa 9633 pana la 1 pentru câteva zeci de clase (9936, 99, etc.). Datasetul a fost împărțit in 3 parți train(70%), validare(15%) si test(15%).

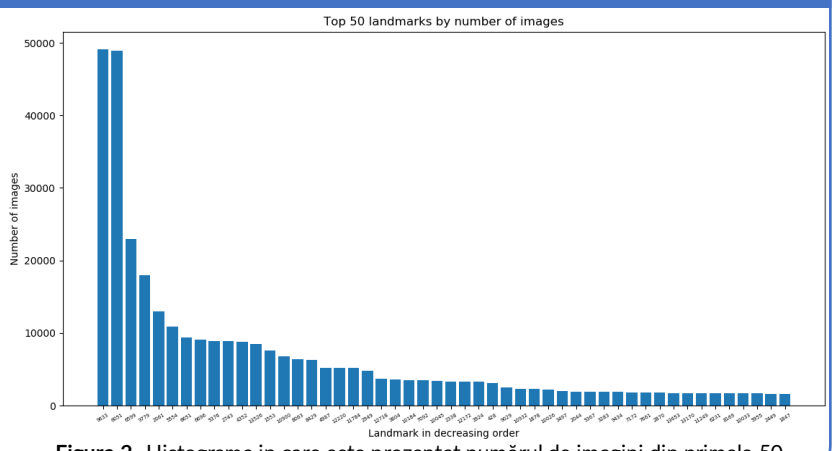

Figura 2- Histograma in care este prezentat numărul de imagini din primele 50 de clase sortate după greutate.

Modulul de atenție

- In continuare extindem rețeaua anterioara la un **Attention Branch Network (ABN)**[2] ceea ce nu este altceva decât VGG-16 cu structura de **branch** si un modul de atenție. Arhitectura de tip branch este sugerata abia după aplicarea funcției ReLU de după cele doua convolutionale 7x7x512 cu filtru (3x3). Scopul **modulului de atenție** este de a forța rețeaua sa se uite numai pe anumite zone din imagine. Menționez ca arhitectura folosita este una similara cu cea descrisa in paperul - **Attention Branch Network: Learning of Attention Mechanism for Visual Explanation[2].**
- **Attention map**-ul se obține pe baza a patru convoluții din **feature map**, reducând numărul de canale de la 512 la numărul de clase (la noi 27) apoi prin aplicarea **sigmoidei**. Acesta este o harta 7x7 in cazul nostru de valori intre 0 si 1, in care zonele cu scor aproape de 1 prezinta interes si cele aproape de 0 nu.
- Prin înmulțirea fiecărui canal din feature map cu attention map-ul se obține un nou feature map. **Loss**-ul este calculat ca o **combinație liniara** dintre partea de jos (attention map + clasificare) si partea de sus (**GAP** (funcționează pe ideea de **estompare** a informației)).

$$Loss_{final} = \alpha \cdot Loss_{Attention\ brach} + \beta \cdot Loss_{Perception\ brach}$$

- α si β sunt **hiperparametrii** care măsoară cat valorează **atenția** (partea de sus) si cat **percepția** (partea de jos).

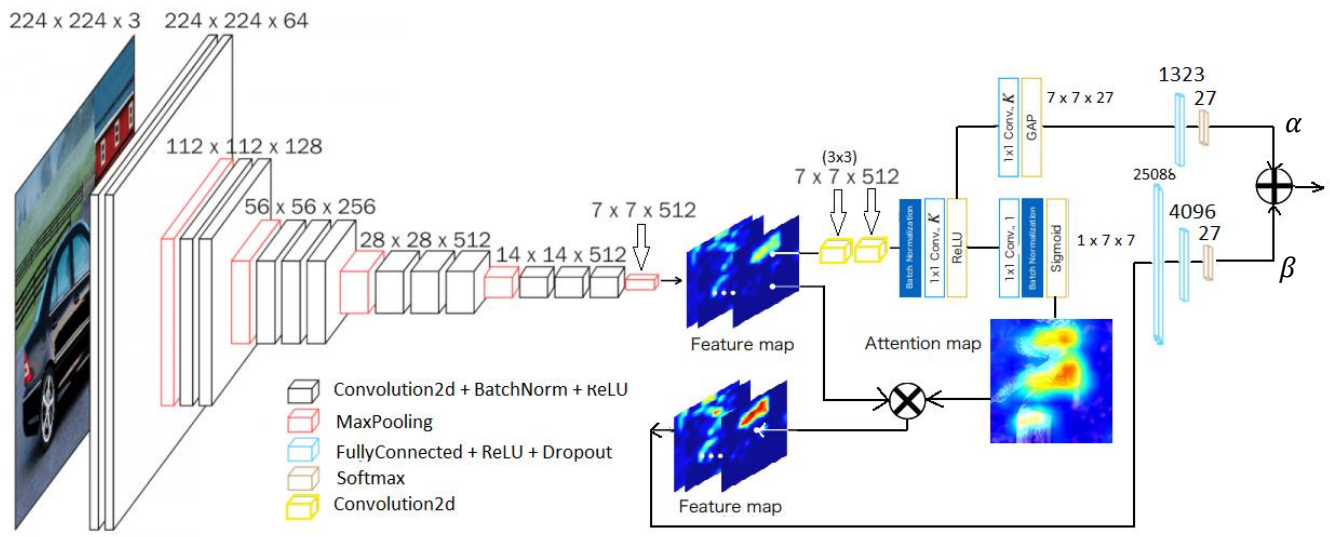

Figura 4 - Arhitectura de VGG-16 modificata peste care s-a adăugat modulul de atenție (overview asupra Attention Branch Network) [2]

Experimente si Rezultate

Modelul	Validare	Test
VGG-16	87.92%	87.97%
VGG-16 cu modul de atenție $\alpha = 0.25, \beta = 1$	30.54%	27.66%
VGG-16 cu modul de atenție $\alpha = 0, \beta = 1$	32.71%	32.58%

Figura 5 - Acuratetea modelelor după 3 epocii pentru diferiți hiperparametrii

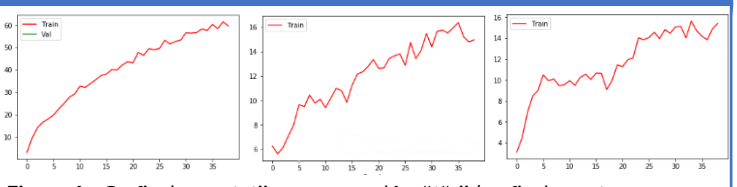

Figura 6 - Graficul acurateții pe parcursul învățării (graficele sunt prezentate pe parcursul epocii 1). Stânga - VGG-16, Mijloc - VGG-16 cu modul de atenție ($\alpha = 0.25, \beta = 1$) Dreapta - VGG-16, Mijloc - VGG-16 cu modul de atenție ($\alpha = 0, \beta = 1$)

- In Figura 5 se observa diferența de acuratetea după 3 epoci intre cele 3 modele, un motiv ar fi numărul mai mare de parametrii al ultimelor doua modele.
- Un aspect interesant este dat de ultimele două linii din table care sugerează faptul ca rețeaua care are lossul format numai de pe ramura de clasificare (cea de percepția) are rata de convergenta puțin mai mare comparativ cu rețeaua care a cumulat lossul din ambele parți, sugerând astfel ca informația estompata (generala) nu aduce un plus in cazul nostru de clasificare.

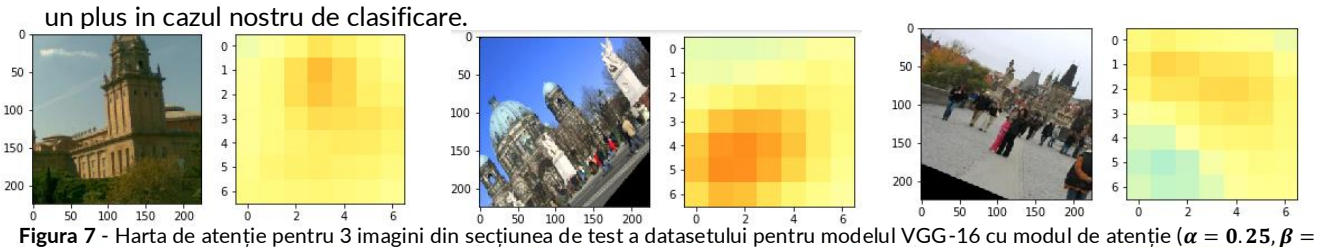

Figura 7 - Harta de atenție pentru 3 imagini din secțiunea de test a datasetului pentru modelul VGG-16 cu modul de atenție ($\alpha = 0.25, \beta = 1$)

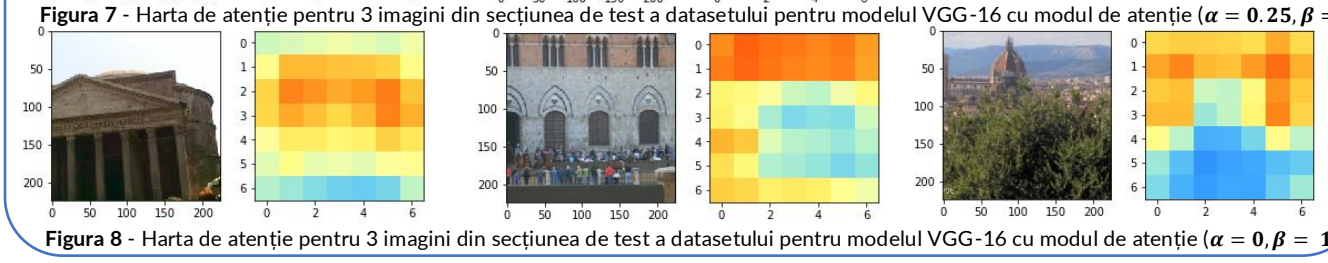

Figura 8 - Harta de atenție pentru 3 imagini din secțiunea de test a datasetului pentru modelul VGG-16 cu modul de atenție ($\alpha = 0, \beta = 1$)

Probleme la implementare

- Una din marile impedimente de care am data la implementare a fost eroarea *CUDA out of memory*, atunci când trecem de la o epoca la alta. Soluție: antrenez doar cate o epoca apoi salvez modelul pe drive. La sfârșit de epoca încarc noii parametrii si repet procedeul pana termin numărul de epoci.
- Un alt impediment este durata mare de train, aproximativ 60-65 minute in acest moment. Menționez ca a fost redus de la aproximativ 3h.

Concluzii

- Atenția ajunge sa se concentreze pe elementele de interes din imagine, însă învățarea este îngreunata de modelul mai complex si necesita o tunare mai atenta a hiperparametrilor si o atenție mai mare la procesul de optimizare
- Modulul de atenție este un concept general care poate fi aplicat la orice task ce presupune modele convolutionale si poate fi aplicat la diverse nivele din rețea, aspect ce nu a apucat sa fie tratat in aceasta lucrare. (fiind pus la final de feature extraction)

Referințe

[1] - Attention Branch Network: Learning of Attention Mechanism for Visual Explanation - Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi
[2] - Very Deep Convolutional Networks for Large-Scale Image Recognition - Karen Simonyan, Andrew Zisserman