

Forest Fire Detection

Firefighters Team – AI Models in Climate Change

1. Introducere

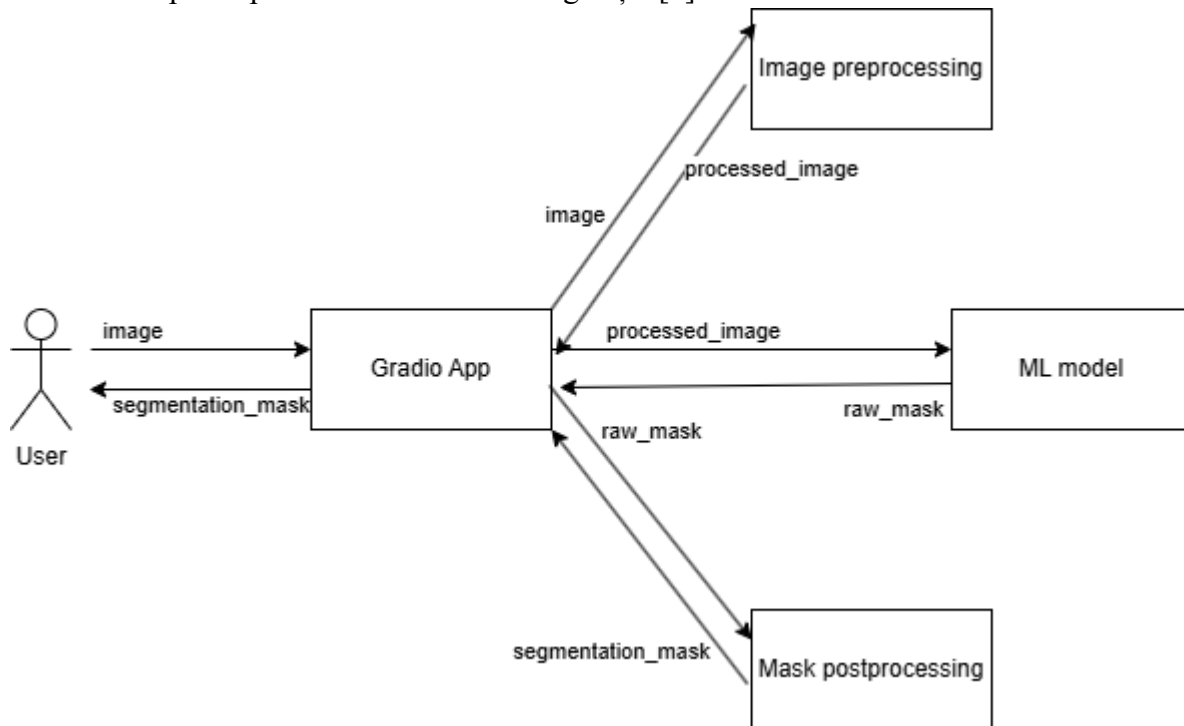
Pădurile joacă un rol crucial în păstrarea echilibrului biologic al planetei, iar incendiile forestiere pot fi catastrofice, având impacte severe și diverse la nivelul mediului înconjurător, al economiei și al vieților omenești. Printre cele mai serioase consecințe ale incendiilor se numără încălzirea globală, pierderea biodiversității și deteriorarea habitatului natural. Predicția și detectarea incendiilor forestiere într-un stadiu incipient sunt vitale pentru atenuarea daunelor provocate de incendii și minimizarea acțiunilor necesare pentru stingerea flăcărilor. Primul pas în prevenirea răspândirii incendiilor forestiere este de a **estima probabilitatea dezvoltării** lor folosind modele care iau în considerare vremea și disponibilitatea combustibilului, aceste modele fiind cruciale în prevenirea aprinderii și răspândirii incendiilor. **Detectarea incendiilor forestiere** este a doua strategie de prevenire a incendiilor și urmărește localizarea rapidă și identificarea flăcărilor, furnizând alerte de incendiu precise pentru a preveni răspândirea acestora [1].

Incendiile de pădure sunt dezastre naturale ce reprezintă o amenințare majoră pentru mediul înconjurător, comunități și ecosistem. În efortul de a atenua riscurile asociate cu astfel de evenimente, inteligența artificială a fost aplicată în domenii diverse pentru prevenirea incendiilor forestiere, detectarea și restaurarea acestora, fiind utilizate tehnici variate pentru modelarea incendiilor. Una dintre cele mai populare tehnici utilizate este învățarea automată (ML - Machine Learning). Algoritmii de învățare automată au potențialul de a recunoaște șabloane și corelații în cadrul datelor, informații care pot fi valorificate pentru a prognoza posibilitatea apariției incendiilor forestiere, a monitoriza progresul expansiunii flăcărilor și a descoperi vulnerabilitatea anumitor regiuni la incendii. Cei mai cunoscuți algoritmi de ML utilizați în modelarea incendiilor sunt Random Forests (Pădure de arbori decizionali), Support Vector Machines (Mașini cu suport vectorial) și Decision Trees (Arbori de decizie) [1].

O altă tehnică de inteligență artificială care a fost aplicată în modelarea incendiilor sunt rețelele neuronale artificiale (ANN). ANN-urile au fost folosite la modelarea comportamentului răspândirii incendiilor, la prezicerea severității și la identificarea surselor potențiale de aprindere. De asemenea, rețelele neuronale artificiale sunt utile și în procesarea cantităților mari de date din diverse surse, cum ar fi imagini din satelit și prognoze meteo, oferind predicții în timp real [1].

Învățarea profundă (Deep Learning - DL) este o ramură a ML care a devenit din ce în ce mai populară în ultimii ani datorită capacității de a învăța și de a se îmbunătăți automat din cantități mari de date. În domeniul modelării incendiilor forestiere, tehnicile DL au fost folosite pentru analiza imaginilor și videoclipurilor, prognoza extinderii incendiilor și evaluarea severității acestora. Rețele neuronale convoluționale (CNN) sunt una dintre cele mai frecvent utilizate metode DL în acest domeniu, deoarece pot învăța automat caracteristici din imagini din satelit, fotografii aeriene sau imagini termice, putând utiliza aceste caracteristici pentru a efectua diverse sarcini, cum ar fi detectarea și segmentarea incendiilor [1].

Utilizarea tehnicilor de AI, ML și DL în prevenirea, detectarea și restaurarea incendiilor forestiere poate îmbunătăți considerabil acuratețea și eficiența predicțiilor, ducând la îmbunătățirea managementului acțiunilor de răspuns și la optimizarea procesului decizional în gestionarea incendiilor forestiere. Combinația diferitelor tehnici de AI, inclusiv cele de DL, pot aborda complexitatea și incertitudinea sistemelor de incendii forestiere, fiind un instrument valoros în lupta împotriva incendiilor de vegetație [1].



2. Metode existente de rezolvare a problemei

O primă lucrare analizată pentru detectarea incendiilor forestiere este [2]. Datele de intrare sunt în format de imagini satelitare cu păduri, autorii nu menționează direct un set de date, doar spun că folosesc “Google Images API” pentru a face rost de imagini, ceea ce ne face să înțelegem că setul de date a fost creat manual. Autorii nu ne oferă setul de date în vreun format, deci setul de date folosit de aceștia nu este disponibil, iar ca date de ieșire avem masca de segmentare unde este posibil să existe foc. Modelul mai poate prezice și dimensiunea în km^2 și nivelul total de dioxid de carbon emis, dar acest lucru posibil să fie scos din masca de segmentare (nu este menționat).

La nivel arhitectural sunt utilizați doi algoritmi: Faster R-CNN (object detection model, cu mențiunea că se poate folosi SSMD - single shot multibox detector în cazul în care Faster R-CNN eșuează): folosit pentru “fire prediction result”, în care sunt detectate zonele în care ar fi foc în imagini și SVM (mașină cu suport vectorial) pentru prezicerea dacă este un foc real sau nu. În lucrare nu se menționează dimensiunea modelului/modul de hypertune a parametrilor, însă arhitectura modelului este descrisă în imaginea [Figure 1]:

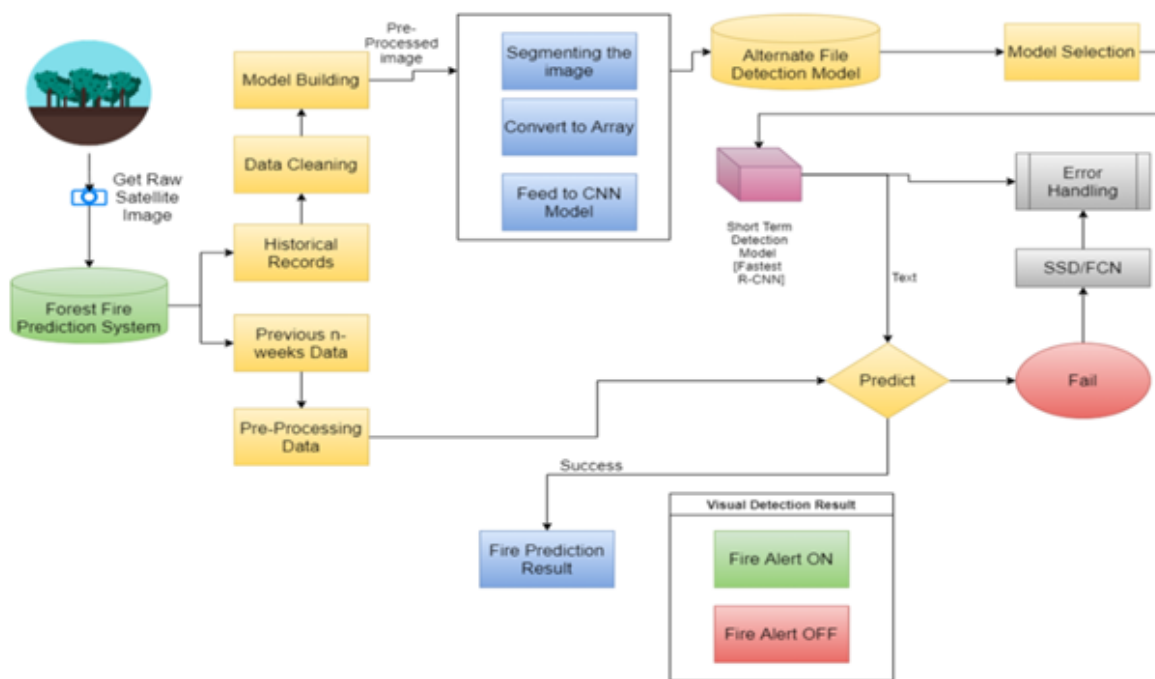


Figure 1 - Arhitectură model, [2]

Antrenarea modelului este sintetizată în figura [Figure 2], iar pentru testarea performanței s-au măsurat metricile acuratețe, precizie și recall, precum și matricea de confuzie. Rezultatele obținute sunt:

- Matricea de confuzie: pentru a identifica unde se “încurcă” modelul.
- Acuratețe: $(\text{total right predictions} / \text{total predictions}) * 100$, rezultat: 92%
- Recall: $TP / (TP + FN)$, rezultat: 97.5%
- Precizie: $TP / (TP + FP)$, rezultat: 84.78%
- F-Measure: $(2 * \text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$, rezultat: 90.7%

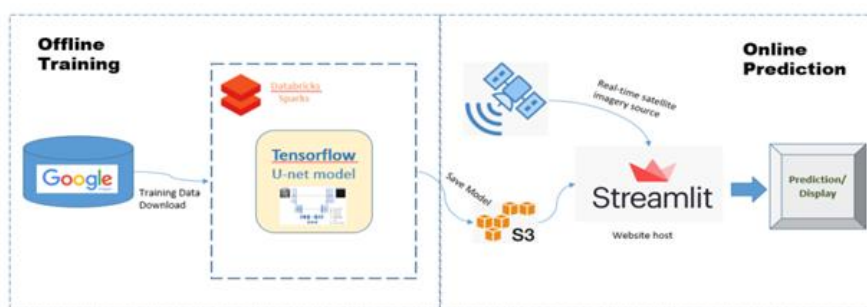


Figure 2 - Antrenarea modelului, [2]

Cel de-al doilea studiu analizat, [3], folosește ca set de date imagini satelitare ce conțin incendii active. Setul de date este obținut de la *NASA Worldview, Earth Observing System Data and Information System* (EOSDIS), instrumentele *Moderate Resolution Imaging Spectroradiometer* (MODIS) pe sateliții Aqusa și Terra și imagini satelitare de la Google: 534 de imagini dintre care 239 au incendii (fire class) și 295 nu (non-fire class); 481 de imagini folosite pentru antrenare (training) și 53 pentru testare (testing). Setul de date conține imagini de satelit dar și alte imagini care au textură similară flăcărilor (focului): lumina soarelui sau

luminozitate în diferite locuri. La nivel arhitectural, pentru clasificarea imaginilor satelitare în imagini cu foc sau fără foc se folosește un model bazat pe rețele neuronale convoluționale numit Inception-v3, iar în plus se mai folosește și algoritmul *Local Binary Pattern* pentru a identifica și regiunile în care apare focul în imagini. Întreaga arhitectură este ilustrată în Figure 3.

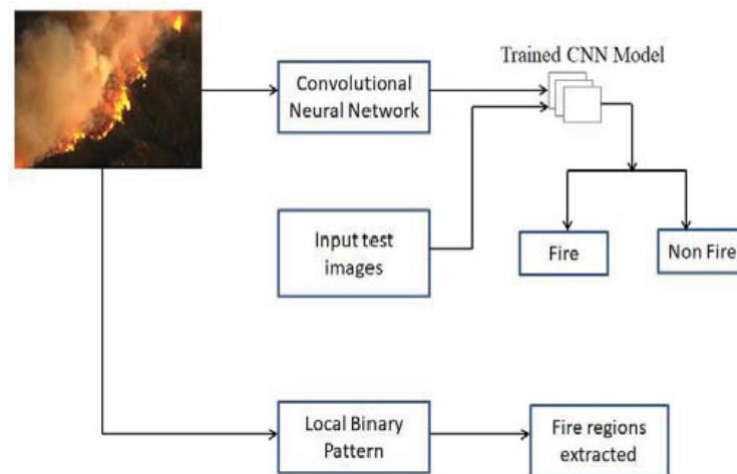


Figure 3 - Arhitectura modelului, [3]

Modelul bazat pe arhitectura Inception-v3 [Figure 4Error! Reference source not found.] rezolvă două sarcini: extragerea de caracteristici din imagini și clasificarea imaginilor. Pentru a **extrage caracteristici** din imagini, modelul CNN este antrenat pe imagini etichetate ca “fire” și “non-fire”, având ca obiectiv identificarea corectă a pixelilor ce conțin foc prin iterații forward și backward multiple care încearcă să reducă funcția de loss *binary cross-entropy*. Ca și input, modelul primește imagini de dimensiuni 299x299. În primul strat de convoluție sunt aplicate 64 de filtre de dimensiune 3×3, generând 64 de hărți de caracteristici. Framework-ul Inception v3 selectează apoi cele mai mari 64 de hărți de caracteristici de activare maximă prin primul strat de *max pooling*, cu un pas de trei pixeli (*stride*), folosind o vecinătate de 3×3 pixeli. Apoi, în al doilea strat de convoluție sunt aplicate 192 de filtre de dimensiune 3×3 imaginii de intrare care generează 192 de hărți de caracteristici. Modelul mai are un singur strat complet conectat cu 448 de filtre de dimensiune 1×1 aplicate imaginii de intrare. **Clasificarea** imaginilor se face prin intermediul unui strat fully-connected (FC) interconectat la sfârșitul rețelei neuronale.

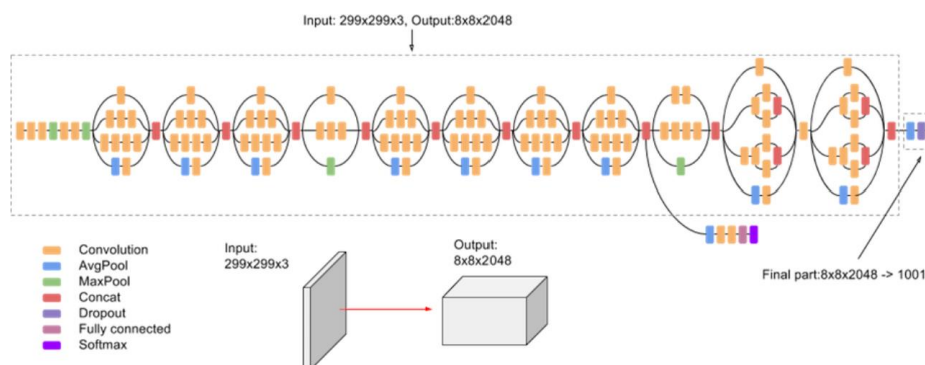


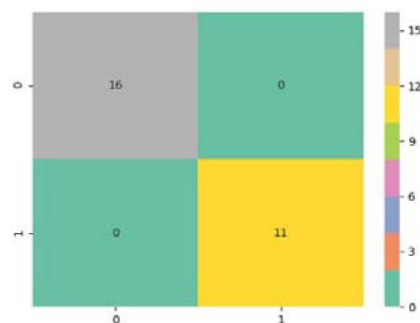
Figure 4 - Arhitectura Inception V3

Pentru detectarea regiunilor cu foc din imagini este aplicat algoritmul *Local Binary Pattern*. Pentru detectarea focului, pentru fiecare pixel se consideră pixelii vecini dintr-o regiune circulară sau rectangulară, urmată de aplicarea unei valori de prag pentru găsirea regiunilor cu foc. Întreaga imagine este procesată folosind un filtru Gaussian Blur cu dimensiunea de intrare de 21×21 , cu limita inferioară de dimensiune $18 \times 50 \times 50$ și limita superioară de dimensiune $35 \times 255 \times 255$. După detectarea regiunii cu foc din masca imaginii de intrare detectată, se aplică un chenar de încadrare în regiunea de foc (obținerea măștii), iar în final, se multiplică imaginea de intrare și masca pentru a obține regiunea afectată de foc.

Metrici calculate sunt precision, recall și F1-Score:

	Precision	Recall	F1-Score
Fire	0.95	1.00	0.97
Non-Fire	1.00	0.97	0.99
Average	0.97	98.50	0.98

Matricea de confuzie



Ultima lucrare analizată [4] prezintă o abordare diferită pentru prezicerea incendiilor forestiere și nu folosește algoritmi pentru procesarea imaginilor, fiind bazat pe mai mulți factori meteorologici, temporali și spațiali. Setul de date utilizat conține 13 astfel de variabile de intrare colectate din Montesinho Natural Park din Portugalia (<https://archive.ics.uci.edu/dataset/162/forest+fires>). Algoritmul folosit se numește rețeaua TOB (Transparent Open Box) și este o metodă bazată pe învățare automată folosită pentru a prezice suprafața arsă în funcție de variabile forestiere, meteorologice și de mediu, evitând utilizarea regresiiilor, corelațiilor și a relațiilor statistice complexe între variabile. Algoritmul presupune două etape (1 și 2) și furnizează două estimări, a doua fiind optimizată. Compararea estimărilor în două etape cu subseturile de date de antrenare și testare ajută la identificarea și respingerea soluțiilor optimizate care se adaptează prea bine variabilelor de date subadiceante (evită overfitting-ul). În prima etapă, TOB stabilește cele mai bune ($Q \leq 10$) potriviri de înregistrări de date într-un subset mare de date de antrenare (pentru fiecare înregistrare de date specifică în subseturi de date relativ mici de ajustare/reglaj - între aproximativ 100 și 150 de înregistrări de date, bazate pe analiza sensibilității). Se compară sumele diferențelor de pătrate ale variabilelor (VSD – variable squared differences) pentru toate variabilele de intrare între înregistrările de date specifice din subsetul de ajustare cu toate înregistrările de date din baza

de date mai mare a subsetului de date de antrenare. În a doua etapă, greutatea variabilelor W_n sunt permise să varieze astfel încât $0 < W_n \leq 1$ și Q este de asemenea permis să varieze ($Q \leq 10$), permițând optimizatorului să minimizeze suma VSD pentru fiecare înregistrare de date de reglaj. Formula genrală de calcul pentru VSD este:

$$\sum VSD_{jk} = \sum_{n=1}^N (X_n)_{jk}^2 W_n$$

Unde N este numărul de variabile de intrare, X_n este valoarea normalizată a variabilei n , j este a j -a înregistrare din subsetul de reglaj, k este a k -a înregistrare din subsetul de antrenament, iar W_n este greutatea aplicată variabilei n în toate comparațiile de înregistrări de date.

Metricile calculate pentru măsurarea acurateții predicției sunt MSE (mean squared error), MAE (mean absolute error), RMSE (root mean squared error), PD (Procent deviation), APD (Average Procent deviation), AAPD (Absolute Average Procent deviation), SD (Standard Deviation) și R (Correlation coefficient) și pot fi vizualizate în [Figure 5].

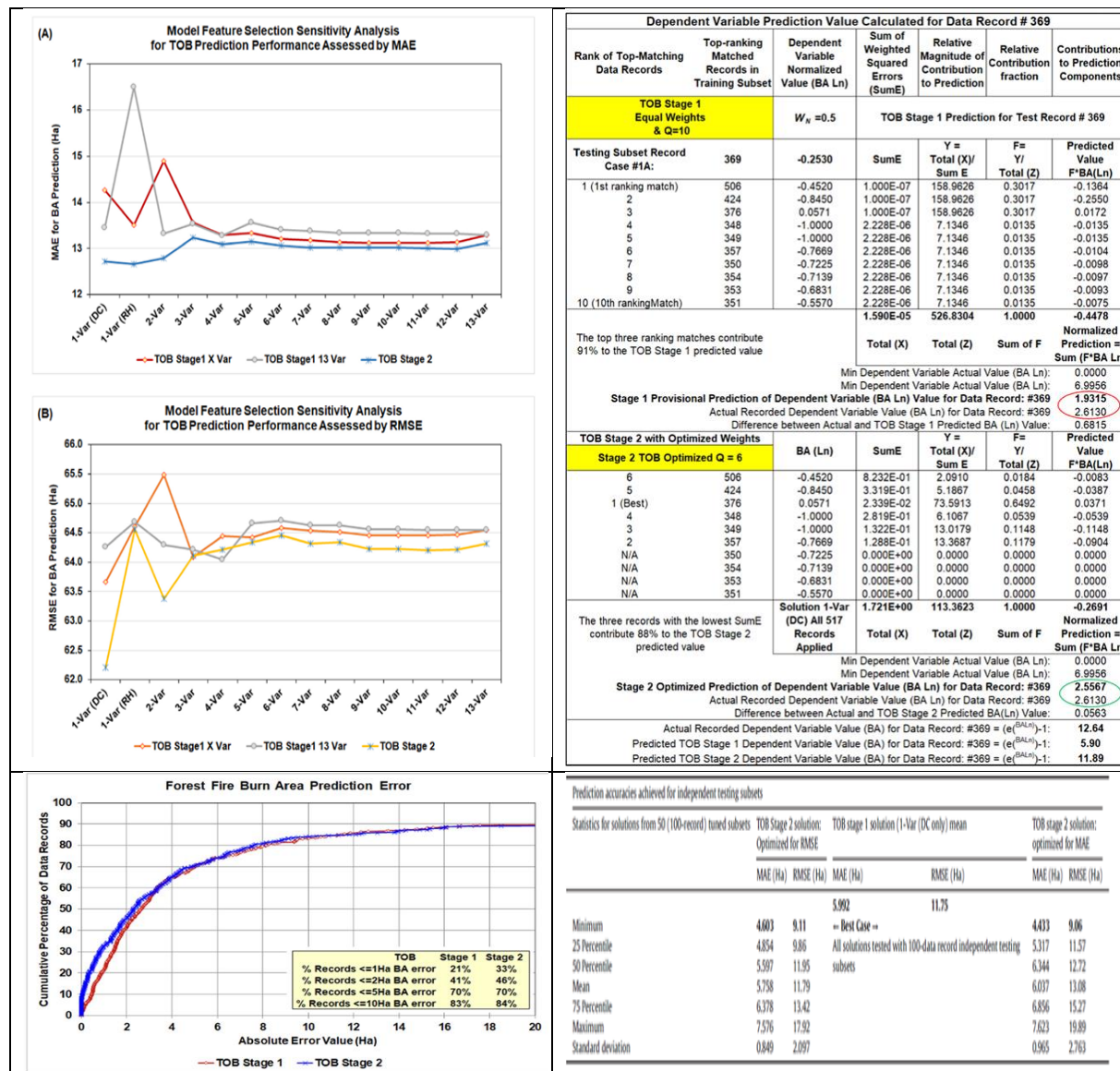


Figure 5 - Metrici calculate, [4]

3. Modele folosite pentru rezolvarea problemei

Pentru rezolvarea problemei de predicție a incendiilor am ales varianta lucrului cu imagini, problema transformându-se într-o problemă de detecție a incendiilor (flăcărilor) din poze. Setul de date utilizat (<https://ieee-dataport.org/open-access/flame-dataset-aerial-imagery-pile-burn-detection-using-drones-uavs>) se numește FLAME (*Fire Luminosity Airborne-based Machine learning Evaluation*) și a fost colectat utilizând drone în timpul unei arderi controlate din Arizona, SUA. În setul de antrenare există aproximativ 39.375 de imagini etichetate în “fire” sau “non-fire”, iar în cel de testare sunt etichetate 8.617 cadre. 2003 cadre sunt considerate pentru segmentarea incendiilor și astfel, 2.003 măști sunt generate în scopul datelor de referință (Ground Truth) cu adnotări la nivel de pixel. Un exemplu de imagine și masca de segmentare corespunzătoare poate fi vizualizat la adresa https://github.com/razvan404/forest-fire-detection/blob/master/data_visualization.ipynb.

Algoritmul propus pentru sarcina de segmentare a focului din imagini este U-Net, o rețea neuronală convolutivă al cărui nume este inspirat de forma de U (adică din 2 componente, partea de feature extraction (extragerea de caracteristici), în care dimensiunile se reduc, dar numărul de canale crește, și partea de mask construction (construirea măștii), în care se folosește informația extrasă și puțin din informația mai veche pentru a asigura că nu se pierd detalii pentru a încerca să construiască masca de segmentare). Arhitectura U-Net este detaliată în figura [Figure 6].

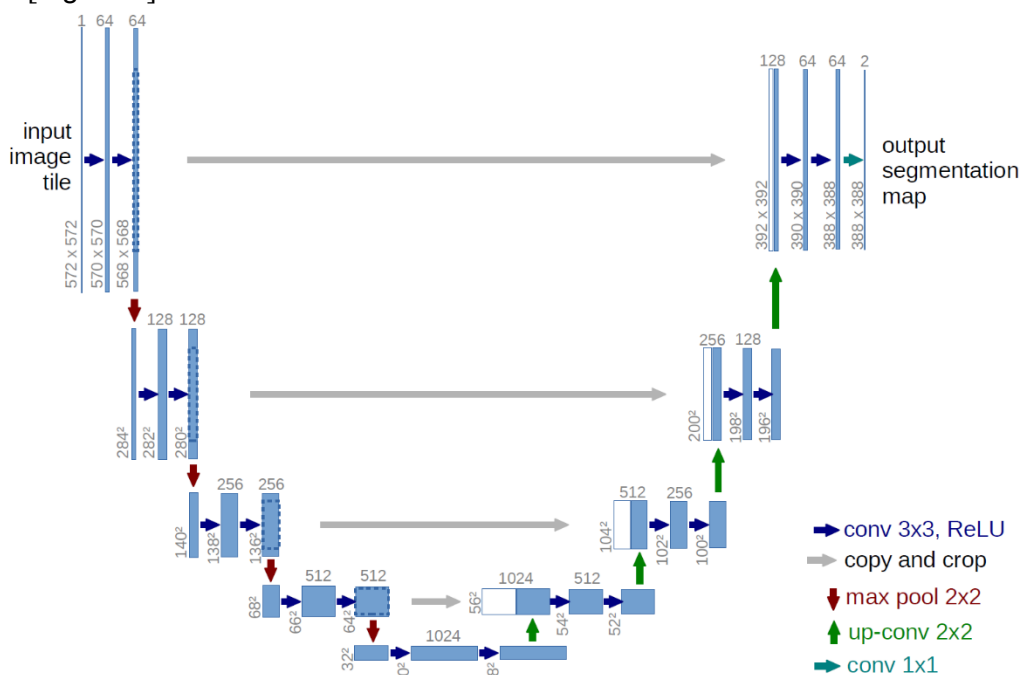


Figure 6 – Arhitectura modelului U-Net [5]

```
=== Run Configuration ===
device = device(type='cuda')
seed = 0
resolution = 128
num_epochs = 8
batch_size = 32
in_channels = 3
out_channels = 1
learning_rate = 1e-05
```

Figure 7 - Parametri inițiali pentru algoritm

Alegerea parametrilor pentru algoritm cuprinde dimensiunea (rezoluția imaginilor) la 128x128 pixeli (a fost selectat central crop-ul imaginii, urmând să fie redimensionată la 128x128 folosind Area Sampling pentru imagine, respectiv Nearest Neighbours pentru măștile de segmentare, dimensiunea a fost aleasă pentru a putea rula experimente cu batch size cât mai mare și un număr cât mai mare de pași, pentru a fi terminat într-un timp rezonabil), numărul de epoci setat la 8, dimensiunea batch-ului la 32, numărul de canale de intrare egal cu 3 (corespunzător

spațiului de culori RGB), cel de ieșire egal cu 1 (în cazul nostru vom avea o mască de segmentare binară, având valorile 1 dacă este foc, respectiv 0 altfel), precum și learning rate inițializat cu valoarea 10^{-5} . Ca și algoritm de optimizare am folosit optimizatorul Adam (*Adaptive Moment Estimation*) pe care l-am setat la valorile de bază (betas: tuple[float, float] = (0.9, 0.999), eps: float = 1e-8).

Valorile metricilor calculate în timpul primului experiment sunt train loss, intersection over union mean (mIoU), pixels accuracy, frequency weighted intersection over union (fwIoU) și validation loss. Rezultatele sunt sumarizate în imaginea [Figure 8], dar pot fi vizualizate detaliat la adresa <https://api.wandb.ai/links/rasvan/xipk15d7>.

4. Rezultate experimentale obținute

Deși experimentul 1 a fost realizat pe o rezoluție mică a imaginilor, se pare că se obțin rezultate destul de bune, atât pe setul de antrenare, cât și pe setul de validare (și se poate observa și din imaginile realizate pe setul de validare că algoritmul pare să detecteze corect locația focului). Se pot încerca mai multe combinații pentru îmbunătățirea performanței. În plus, se mai pare că după 8 epoci algoritmul pare că începe să converge, deci antrenamentul pentru următoarele experimente nu ar trebui să dureze mult mai mult de 12-16 epoci.

Antrenamentul [Figure 9] a fost realizat pe o placă video GPU (GTX 1650Ti, 4GB) și a durat în jur de 1 oră, pentru a realiza mai multe experimente într-un timp mai scurt se poate folosi o unitate de calcul mai puternică. Astfel, se poate crește și numărul de epoci, cât și rezoluția imaginilor sau batch size.

▼ Results

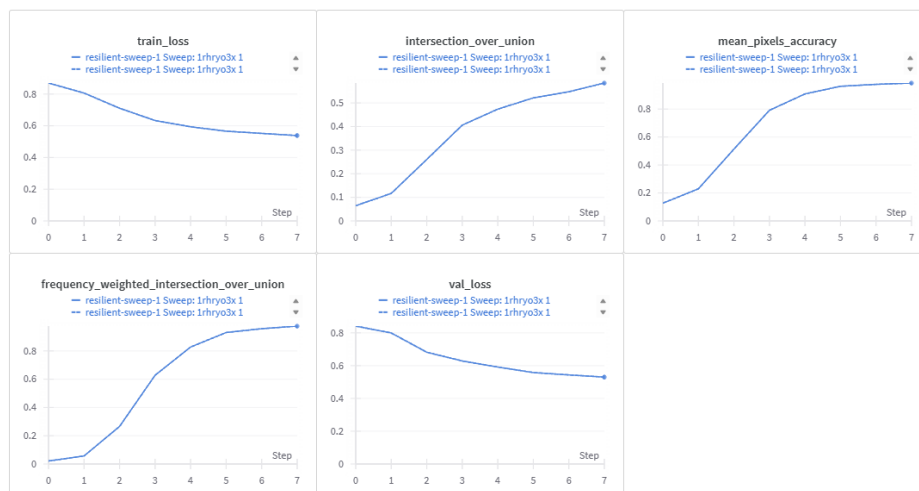


Figure 8 - Valori metrice experiment 1

```
Epoch 0 - Train: 100%|██████████| 51/51 [05:28<00:00, 6.44s/it, loss=0.842]
Epoch 0 - Validation: 100%|██████████| 51/51 [01:24<00:00, 1.66s/it, fwioU=0.0388, loss=0.818, miou=0.0309, mpa=0.0618]

Epoch 0: Train Loss: 0.8696 | Validation Loss: 0.8412 | MPA: 0.1271 | MIoU: 0.0644 | FWIoU: 0.0214

Epoch 1 - Train: 100%|██████████| 51/51 [05:26<00:00, 6.40s/it, loss=0.773]
Epoch 1 - Validation: 100%|██████████| 51/51 [01:27<00:00, 1.72s/it, fwioU=0.0184, loss=0.78, miou=0.0677, mpa=0.135]

Epoch 1: Train Loss: 0.8063 | Validation Loss: 0.7994 | MPA: 0.2297 | MIoU: 0.1166 | FWIoU: 0.0569

Epoch 2 - Train: 100%|██████████| 51/51 [05:25<00:00, 6.37s/it, loss=0.657]
Epoch 2 - Validation: 100%|██████████| 51/51 [01:27<00:00, 1.72s/it, fwioU=0.214, loss=0.705, miou=0.231, mpa=0.463]

Epoch 2: Train Loss: 0.7106 | Validation Loss: 0.6819 | MPA: 0.5134 | MIoU: 0.2604 | FWIoU: 0.2660

Epoch 3 - Train: 100%|██████████| 51/51 [05:19<00:00, 6.26s/it, loss=0.702]
Epoch 3 - Validation: 100%|██████████| 51/51 [01:25<00:00, 1.67s/it, fwioU=0.574, loss=0.651, miou=0.379, mpa=0.758]

Epoch 3: Train Loss: 0.6330 | Validation Loss: 0.6284 | MPA: 0.7911 | MIoU: 0.4060 | FWIoU: 0.6280

Epoch 4 - Train: 100%|██████████| 51/51 [05:19<00:00, 6.27s/it, loss=0.64]
Epoch 4 - Validation: 100%|██████████| 51/51 [01:25<00:00, 1.67s/it, fwioU=0.809, loss=0.612, miou=0.45, mpa=0.9]

Epoch 4: Train Loss: 0.5937 | Validation Loss: 0.5916 | MPA: 0.9081 | MIoU: 0.4737 | FWIoU: 0.8276

Epoch 5 - Train: 100%|██████████| 51/51 [05:25<00:00, 6.38s/it, loss=0.563]
Epoch 5 - Validation: 100%|██████████| 51/51 [01:26<00:00, 1.70s/it, fwioU=0.902, loss=0.588, miou=0.475, mpa=0.95]

Epoch 5: Train Loss: 0.5659 | Validation Loss: 0.5578 | MPA: 0.9626 | MIoU: 0.5213 | FWIoU: 0.9302

Epoch 6 - Train: 100%|██████████| 51/51 [05:23<00:00, 6.34s/it, loss=0.768]
Epoch 6 - Validation: 100%|██████████| 51/51 [01:25<00:00, 1.67s/it, fwioU=0.94, loss=0.576, miou=0.485, mpa=0.97]

Epoch 6: Train Loss: 0.5520 | Validation Loss: 0.5433 | MPA: 0.9766 | MIoU: 0.5473 | FWIoU: 0.9573

Epoch 7 - Train: 100%|██████████| 51/51 [05:29<00:00, 6.47s/it, loss=0.561]
Epoch 7 - Validation: 100%|██████████| 51/51 [01:25<00:00, 1.67s/it, fwioU=0.973, loss=0.552, miou=0.493, mpa=0.986]

Epoch 7: Train Loss: 0.5383 | Validation Loss: 0.5301 | MPA: 0.9861 | MIoU: 0.5843 | FWIoU: 0.9758
```

Figure 9 - Modificarea metricilor în timpul antrenamentului

La cel de-al doilea experiment am urmărit modificarea comportamentului metricilor în funcție de modificarea unor parametri ai algoritmului. Ambele experimente au păstrat rezoluția imaginii la dimensiuni de 128x128 pixeli, singurele modificări le-am adus parametrilor *learning rate* și *batch size*. În cadrul experimentului 2, cea mai bună rulare (care a obținut cele mai bune valori ale metricilor de evaluare) s-a obținut pentru *batch_size=8* și *learning_rate*

$=10^{-3}$. S-a observat că la creșterea *batch_size* sau la micșorarea *learning_rate*, performanța algoritmului a scăzut. Comparând rezultatele cu experimentul 1, unde *batch_size*=32 și *learning_rate*= 10^{-5} , se remarcă faptul că toate metricile calculate s-au îmbunătățit, după cum este ilustrat și în Table 1.

Experiment	Train Loss	IoU	MPA	fwIoU	Val Loss
1	0.5383	0.5843	0.9861	0.9758	0.5301
2	0.1415	0.8436	0.9974	0.9952	0.1222

Table 1 - Analiză comparativă a celor doua experimente

Valorile metricilor în timpul antrenării modelului sunt ilustrate [Figure 10], însă un raport mai detaliat al acestora poate fi vizualizat și la <https://api.wandb.ai/links/rasvan/79tt24hc>.

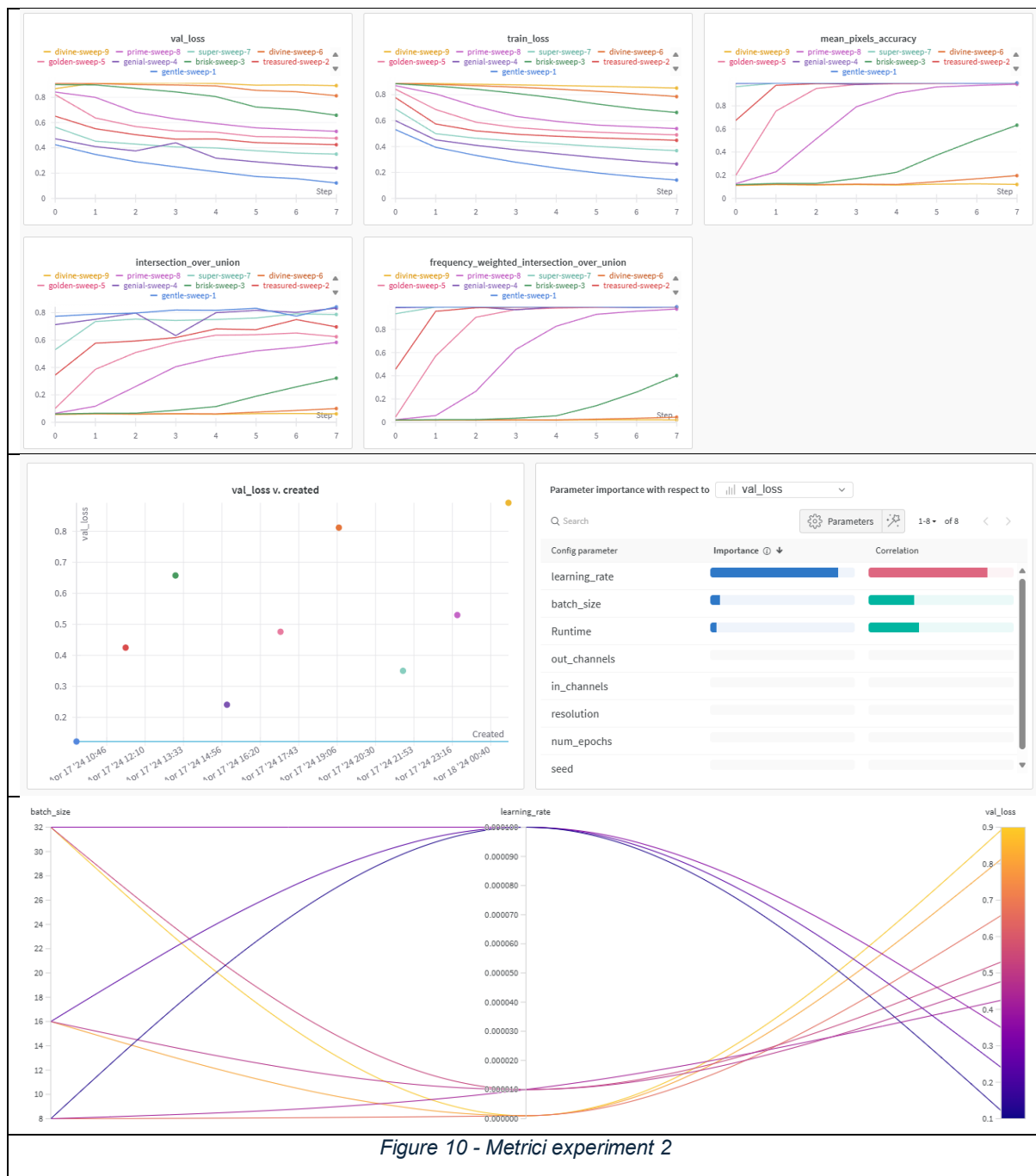


Figure 10 - Metrici experiment 2

5. Concluzii și posibile îmbunătățiri

În cadrul proiectului dezvoltat, am utilizat arhitectura U-Net pentru realizarea segmentării și a detecției incendiilor din imagini. Cele două experimente ne arată importanța alegerii parametrilor potriviți pentru algoritmi în vederea îmbunătățirii performanței acestora. Deși metricile obținute sunt satisfăcătoare, o și posibilă direcție de dezvoltare ar fi utilizarea unei arhitecturi mai performante pentru U-Net precum Attention U-Net și compararea rezultatelor cu cele obținute în cadrul experimentelor, pentru a analiza modul în care se modifică acuratețea obținută.

6. Referințe

- [1] S. Giannakidou, P. Radoglou-Grammatikis, T. Lagkas, V. Argyriou, S. Goudos, E. K. Markakis și P. Sarigiannidis, „Leveraging the power of internet of things and artificial intelligence in forest fire prevention, detection, and restoration: A comprehensive survey,” *Internet of Things*, vol. 26, p. 101171, 2024.
- [2] M. K. S., S. B. J., P. S., R. Sharma, S. Tej și S. A. Karani, „Forest Fire Prediction Using Image Processing And Machine Learning,” *NVEO – Natural Volatiles & Essential Oils*, vol. 8, nr. 4, pp. 13116-13134, 2021.
- [3] P. R. Shanmuga și V. K., „Deep Learning Based Forest Fire Classification and Detection in Satellite Images,” *11th International Conference on Advanced Computing (ICoAC)*, pp. 61-65, 2019.
- [4] D. A. Wood, „Prediction and data mining of burned areas of forest fires: Optimized data matching and mining algorithm provides valuable insight,” *Artificial Intelligence in Agriculture*, vol. 5, pp. 24-42, 2021.
- [5] O. a. F. P. a. B. T. Ronneberger, „U-net: Convolutional networks for biomedical image segmentation,” în *Medical image computing and computer-assisted intervention--MICCAI 2015: 18th international conference, October 5-9, 2015, proceedings, part III 18*, Munich, Germany, 2015.
- [6] Firefighters, „Forest Fire Detection,” [Interactiv]. Available: <https://github.com/razvan404/forest-fire-detection> .

7. Anexe

- Link către repository GIT: <https://github.com/razvan404/forest-fire-detection>