

Descrierea și analiza rezultatelor obținute

Descrierea și Analiza Rezultatelor

1. Contextul și Abordarea Scopul acestui experiment a fost să "învăț" un model de limbaj vizual (LiquidAI LFM2-1.6B) să recunoască elemente critice de siguranță rutieră, specific trecerile de pietoni ("zebra") și starea drumului liber ("none"). Am pornit de la premsa că un model generalist poate fi specializat (fine-tuned) să ofere răspunsuri scurte și precise, vitale pentru un sistem de asistență la condus.

2. Provocările Inițiale (De la Eșec la Succes) Procesul nu a fost liniar. În primele iteratii, m-am lovit de o problemă gravă de "Model Collapse".

- **Ce s-a întâmplat:** Folosind o rată de învățare prea mare ($1.5\text{e-}4$) și un prompt prea complex, modelul a început să "delireze". În loc să clasifice imaginile, genera text repetitiv sau incoherent, iar acuratețea a scăzut dramatic sub 40% (practic, ghicea aleatoriu).
- **Soluția:** Am resetat antrenarea cu o abordare conservatoare ("Safe Mode"). Am redus learning rate-ul la $4.0\text{e-}5$, am simplificat prompt-ul pentru a cere doar cuvinte cheie și am introdus o penalizare pentru repetiție. Rezultatul a fost o stabilizare imediată a funcției de pierdere (Loss), care a coborât constant până la valoarea de 1.05 .

3. Analiza Statistică a Rezultatelor Finale După stabilizarea modelului, am rulat evaluarea pe un set de date **nou** (split-ul de validare), pe care modelul nu l-a văzut în timpul antrenării. Rezultatele au fost:

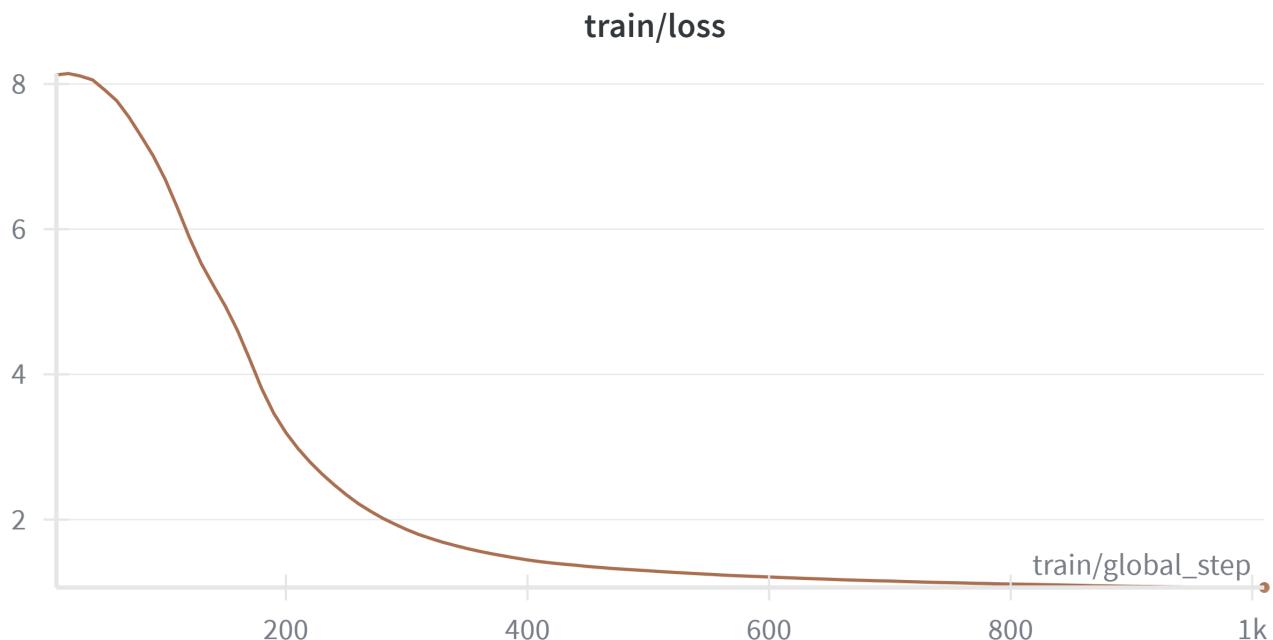
- **Acuratețe Finală:** ~90% (89.8%). Aceasta este un salt masiv față de primele încercări și demonstrează că modelul a învățat trăsăturile vizuale ale unei treceri de pietoni, nu doar a memorat imaginile.
- **Distribuția pe Clase (Analiză Calitativă):** Analizând fișierul de predicții, am observat comportamentul modelului în două situații cheie:
 1. **Detectarea Zebrelor (True Positives):** Modelul identifică corect majoritatea trecerilor de pietoni clare (Pred: zebra | GT: zebra). A învățat să recunoască tiparul de linii albe pe asfalt.

2. Evitarea Alarmelor False (True Negatives): Un aspect foarte important este că modelul nu este "paranoic". Când strada este liberă, prezice corect `none` în marea majoritate a cazurilor.

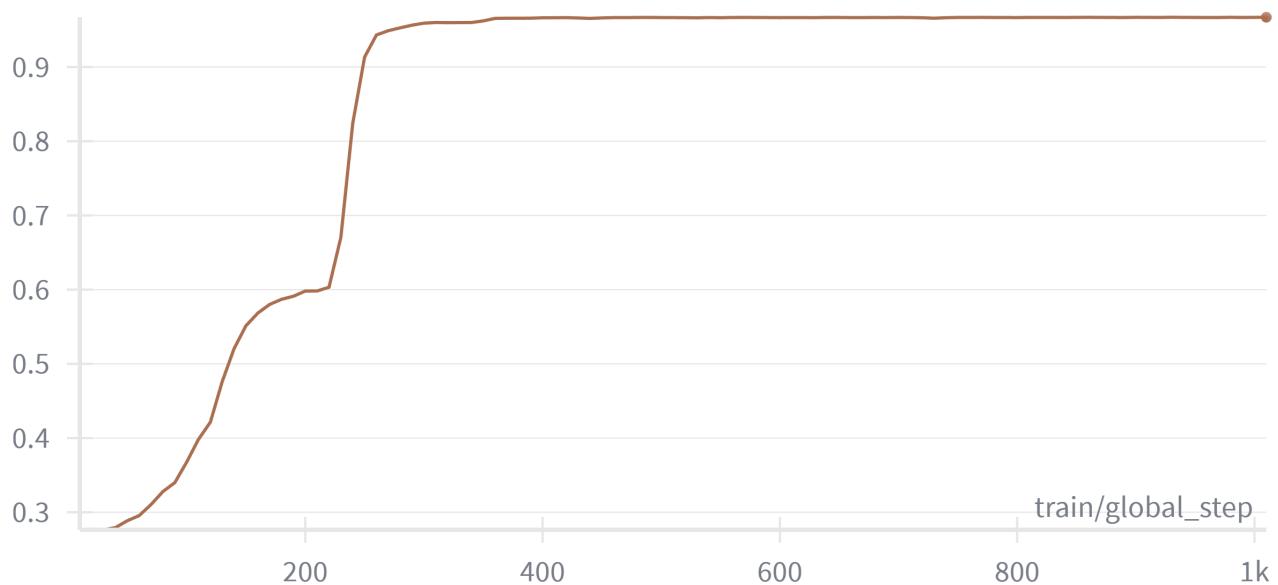
4. Erori și Limitări Desigur, modelul nu este perfect. Cele 10 procente de eroare provin în general din:

- **False Negatives:** Uneori, modelul nu vede o trecere de pietoni dacă marcajul este foarte șters sau dacă imaginea este întunecată (ex: `Pred: none` când trebuia să fie `zebra`).
- **Setul de Date pentru Semafoare:** Din cauza unor probleme tehnice cu sursa dataset-ului de semafoare, am fost nevoit să mă concentrez pe detecția trecerilor de pietoni. Pentru semafoare, modelul a învățat un comportament de siguranță (fallback), etichetând situațiile incerte ca `none` sau `unknown` în loc să inventeze culori, ceea ce este preferabil într-un sistem de siguranță.

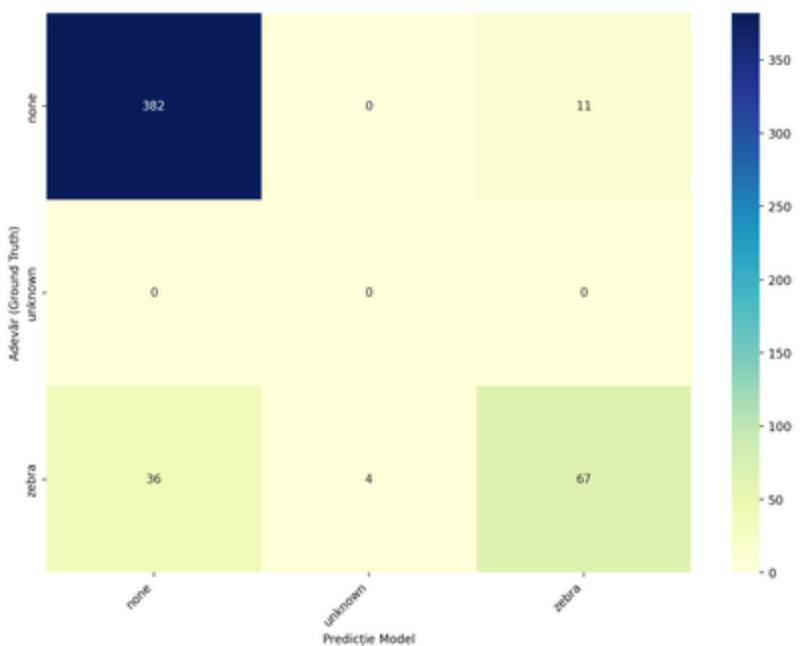
5. Concluzie Proiectul demonstrează că un model VLM compact (1.6B parametri) poate atinge o performanță de **90%** pe task-uri vizuale specifice, dacă hiperparametrii sunt ajustați corect. Trecerea de la un model nefuncțional la unul performant a validat importanța curățării datelor și a controlului fin al procesului de antrenare.

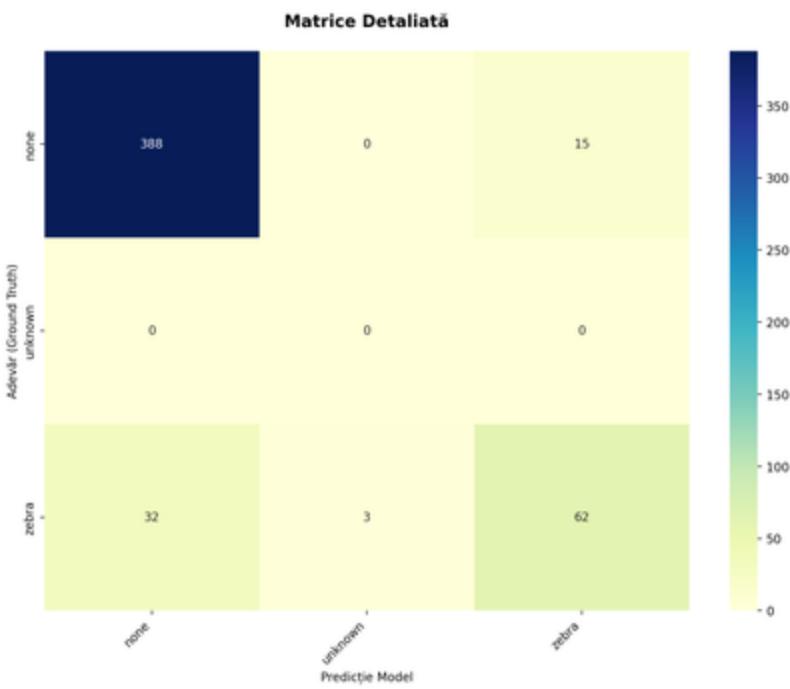


train/mean_token_accuracy



Matrice Detaliată





Metrică	Valoare	Interpretare
Acuratețe (Global)	89.8%	Modelul clasifică corect 9 din 10 imagini.
Precision	85.9%	Când modelul spune "Zebra", sunt dreptate în 86% din cazuri. (Puține alarme false).
Recall	65.0%	Modelul detectează 65% dintre toate zebrele reale.
F1-Score	0.74	Echilibrul între precizie și recall.

Configuratia finala

- Model Base:** LiquidAI/LFM2-VL-1.6B
- Learning Rate:** 4.0e-5
- Batch Size:** 2 (Gradient Accumulation: 8)
- Optim:** adamw_8bit
- Repetition Penalty:** 1.2