ACADEMIA ROMÂNĂ

Școala de Studii Avansate a Academiei Române

Institutul de Matematică „Simion Stoilow”

Rezumatul Tezei de Doctorat

Învățare semi supervizată a multiplelor taskuri folosind grafuri neurale de concepte

DOCTORAND

Pîrvu Mihai-Cristian

CONDUCĂTOR DE DOCTORAT

Prof. Dr. Marius Leordeanu

2025

Contents

1 Introducere **2**

2 Concepte Fundamentale **5**

2.1 Datele. Combustibilul care alimentează Învățarea Automată 5

2.2 Modele. Modelarea distribuțiilor de date cu Învățarea Automată 6

3 Învățarea prin ansamblu și distilarea cunoștințelor metodelor neurale și analitice pentru estimarea adâncimii **15**

3.1 Prezentare generală a distilării adâncimii metrice 16

3.2 Analiză Experimentală 17

3.3 Concluzii 19

4 Consens pe Grafuri Neurale Multi-Strat pentru Învățare Semi-supervizată **20**

4.1 NGC: Modelul de Consens pe Grafuri Neurale 21

4.2 Analiză experimentală 23

4.3 Concluzii 25

5 Hiper-Grafuri Multi-Strat pentru Învățare Semi-Supervizată **26**

5.1 Modelul de Hiper-Graf Multi-Strat 28

5.2 Analiză experimentală pe setul de date Dronescapes 30

5.3 Analiză experimentală pe setul de date NEO 33

5.4 Concluzii 34

6 Hiper-Grafuri Probabilistice folosind Autoencodere Mascate, ansambluri și distilare eficientă **35**

6.1 PHG-MAE: Modelul de Hiper-Grafuri Probabilistice folosind Autoencodere Mascate 37

6.2 Analiză experimentală 39

6.3 Concluzii 43

7 Concluzii și direcții viitoare de cercetare **45**

Chapter 1

Introducere

În ultimii ani, Învățarea Automată s-a schimbat considerabil. Rețelele neurale mari, antrenate cu metode precum Stochastic Gradient Descent (SGD), au devenit standardul pentru lucrul cu seturi de date mari și complexe. Această abordare, cunoscută sub numele de Învățare Adâncă (Deep Learning), a condus la numeroase progrese în domenii precum Viziunea Computerizată și Procesarea Limbajului Natural. Când am început sa lucrez la metodelete prezentate în această teză, observasem deja această trecere de la metodele mai vechi, manuale, la cele noi, bazate pe date. Era clar că învățarea adâncă era foarte puternică, dar era mai puțin clar dacă reprezenta soluția completă pentru crearea de sisteme inteligente.

Studii recente sugerează că simpla mărire a modelelor sau furnizarea mai multor date duce la câștiguri din ce în ce mai mici în performanță. Lucrările teoretice arată, de asemenea, că rețelele neurale actuale au limite fundamentale și nu pot rezolva anumite tipuri de probleme complexe. Percepția mea a fost că învățarea adâncă pură este un instrument foarte important, dar este doar o piesă dintr-un puzzle mai mare. Poate fi comparată cu partea intuitivă, de recunoaștere a tiparelor, a gândirii noastre, care este separată de partea mai structurată, de raționament. Această teză explorează modul în care putem construi pe punctele forte ale învățării adânci, abordând în același timp unele dintre limitările sale.

Întrebări cheie și obiective de cercetare

Pentru a realiza acest lucru, cercetarea mea s-a concentrat pe combinația a cinci domenii cheie: rețele neurale adânci, grafuri, consensul predicțiilor prin ansamblu (*ensemble learning*), învățare iterativă semi-supervizată cu distilarea modelelor și aplicarea practică a înțelegerii imaginilor aeriene. Am ales rețelele neurale ca tehnologie de bază, dar am folosit grafuri pentru a modela relațiile dintre diferiți senzori și tipuri de date, la fel cum noi percepem lumea prin simțuri diferite. Pentru a face predicțiile acestor modele mai fiabile, am folosit învățarea prin ansamblu, unde mai multe modele votează asupra unei decizii finale. Pentru a valorifica la maximum datele disponibile, am folosit învățarea semi-supervizată și distilarea, unde un model antrenat pe o cantitate mică de date labeluite învață un nou model folosind o cantitate mare de date nelabeluite. În final, toată această muncă a fost fundamentată pe problema

reală a înțelegerii imaginilor aeriene, cu scopul de a crea modele suficient de eficiente pentru a fi utilizate pe roboți și drone.

Toate acestea pot fi văzute și în Figura 1.1 de mai jos.

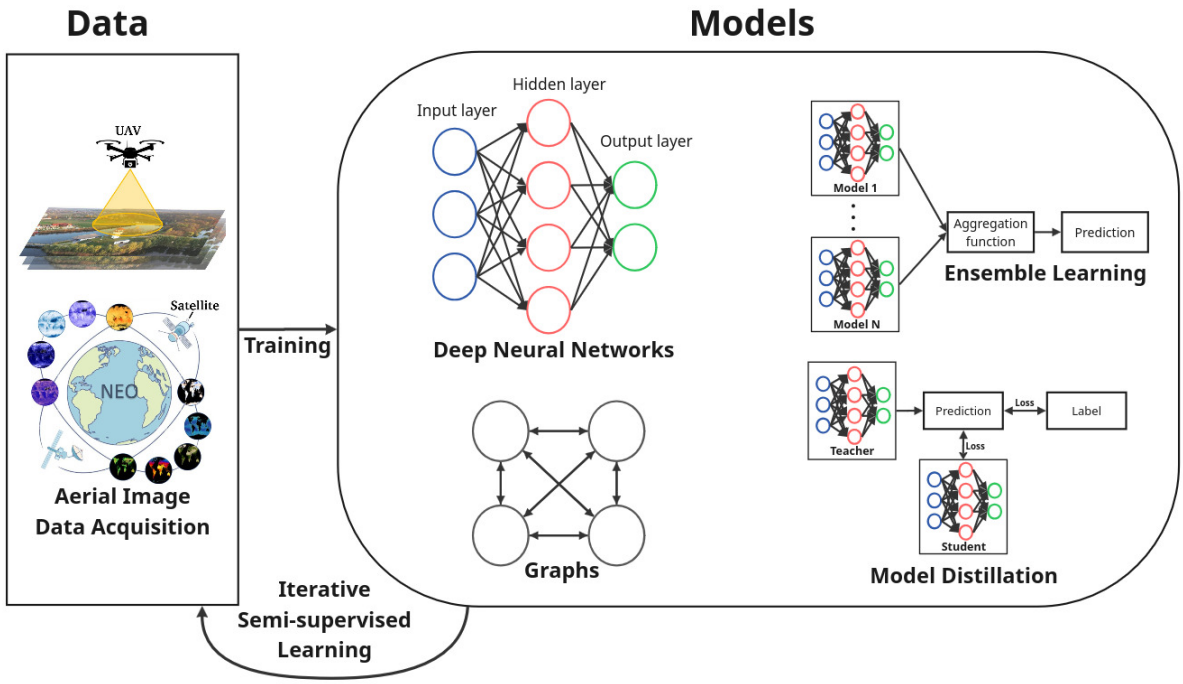


Figure 1.1: Figura principală: blocurile de bază ale acestei teze.

Seturi de date Am introdus mai multe seturi de date noi pentru a ajuta comunitatea de cercetare. Acestea includ un set de date sintetic pentru înțelegerea scenelor aeriene, generat cu simulatorul CARLA, un set de date real numit Dronescapes, cu zboruri deasupra peisajelor românești și norvegiene, și un set de date agregat de la NASA Earth Observations pentru prezicerea măsurătorilor legate de climă. Prezentăm, de asemenea, o extensie a Dronescapes și un cadru open-source pentru a ajuta pe alții să-și creeze propriile seturi de date multi-modale din videoclipuri.

Metode și Algoritmi Contribuțiile metodologice se concentrează pe combinarea ideilor de bază prezentate anterior.

- Am dezvoltat noi metode pentru estimarea adâncimii prin combinarea abordărilor analitice și a celor bazate pe rețele neurale, folosind ansambluri.
- Am introdus Hiper-Grafurile Neurale de Concepte ca o modalitate de a realiza învățare multi-modală și multi-task, unde fiecare parte a grafului reprezintă o perspectivă diferită asupra lumii.
- Am propus și validat metode de antrenament iterativ semi-supervizat care utilizează grafuri, ansambluri și distilare, oferind dovezi atât teoretice, cât și practice ale eficacității lor pe seturi de date aeriene mari.

- Am arătat că metodele noastre produc predicții mai consistente și coerente temporal, chiar și atunci când sunt antrenate doar pe imagini statice.
- Am dezvoltat o abordare eficientă de distilare pentru a antrena modele simple și rapide din modele mai complexe, făcându-le potrivite pentru utilizare în timp real pe hardware cu resurse limitate.
- Am prezentat o implementare simplificată și mai eficientă a conceptului de Hiper-Graf, arătând că acesta poate fi reprezentat de o singură rețea neurală, ceea ce accelerează semnificativ antrenamentul și inferența.

Restul tezei este structurat pentru a construi pe aceste idei pas cu pas. Vom acoperi mai întâi conceptele de bază, apoi vom prezenta munca noastră privind estimarea adâncimii, urmată de dezvoltarea modelelor noastre bazate pe grafuri atât pe date sintetice, cât și pe date reale, și vom încheia cu modelul nostru unificat și direcțiile viitoare.

Chapter 2

Concepte Fundamentale

Acest capitol oferă o introducere generală a conceptelor tehnice utilizate în teză. Scopul este de a defini termenii și tehnologiile standard pe care le vom folosi în capitolele următoare. Vom începe cu datele pentru UAV-uri și învățarea multi-task în Secțiunea 2.1. Apoi, în Secțiunea 2.2, vom introduce concepte de bază despre Învățarea Automată și Rețele Neurale, explicând cum sunt definite, antrenate și aplicate în probleme de învățare adâncă. Vom discuta, de asemenea, despre învățarea multi-task și multi-modală și vom încheia capitolul cu o scurtă introducere în grafuri, deoarece acestea sunt utilizate pe larg în capitolele ulterioare.

2.1 Datele. Combustibilul care alimentează Învățarea Automată

Lucrarea noastră din [19] a introdus un set de date sintetic cu hărți de adâncime și de „aterizare sigură". Recent, seturile de date mari de tip viziune-limbaj au devenit, de asemenea, populare, cu miliarde de imagini cu descrieri [28].

Setul de date SafeUAV

În lucrarea noastră SafeUAV [19], am folosit Google Earth [12] pentru a crea automat imagini sintetice de tip UAV cu labeluri. Ideea a fost să folosim reconstrucții 3D ale unor locuri reale pentru a ne apropia cât mai mult de scenariile de zbor reale. Deși imaginile nu erau perfect realiste, au fost utile pentru antrenarea modelelor care puteau transfera o parte din ceea ce au învățat pe imagini din lumea reală.

Setul de date are 11.907 eșantioane, împărțite în seturi de antrenament (80%) și validare (20%). Am colectat date din patru zone diferite — două urbane și două suburbane — pentru a diversifica setul de date. Pentru fiecare eșantion, avem o imagine RGB de 640x480, date de adâncime per-pixel și o hartă semantică ce labeluiește suprafețele ca fiind orizontale, verticale sau altele (HVO).

Scenă	Suprafață	Antrenament	Validare
Suburban A	1.7km^2	1966	492
Suburban B	1.1km^2	1049	263
Urban A	3.5km^2	3636	909
Urban B	3.3km^2	2873	819

Table 2.1: Distribuția scenelor în setul de date SafeUAV.

Scopul lucrării originale a fost de a găsi zone sigure de aterizare din imagini RGB. Am formulat aceasta ca un task de segmentare semantică la nivel de pixel cu trei clase: „orizontale" (sigure pentru aterizare), „verticale" (obstacole de evitat) și „altele" (suprafețe înclinate sau neregulate). Deși aceasta este o viziune simplificată a lumii (de exemplu, o suprafață de apă orizontală nu este sigură), oferă o bună înțelegere geometrică a scenei doar dintr-un flux video de la cameră. Figura 2.1 arată un eșantion cu labelurile corespunzătoare HVO și de adâncime.

Figure 2.1: Eșantion din setul de date SafeUAV. Coloane: RGB, HVO și adâncime metrică

2.2 Modele. Modelarea distribuțiilor de date cu Învățarea Automată

Învățarea Automată a devenit un instrument standard pentru modelarea datelor. Abordarea s-a schimbat de-a lungul anilor. La început, experții creau manual feature-uri din date, care erau apoi introduse într-un algoritm de învățare simplu. După 2012, modelele end-to-end care învață feature-urile direct din datele brute au devenit populare. Această abordare bazată pe date, susținută de seturi de date mari și hardware mai bun, s-a dovedit a fi mult mai de succes. Astăzi, accentul se pune pe construirea de seturi de date masive și utilizarea metodelor auto-supervizate, deși există un interes crescând pentru combinarea învățării bazate pe date cu cunoștințe fundamentale de domeniu.

Probleme clasice în Învățarea Automată

În esență, Învățarea Automată se referă la învățarea tiparelor din date. Un set de date conține diferite variabile sau feature-uri. Dacă o variabilă este un număr real (precum temperatura), este o problemă de regresie. Dacă este una dintr-un set fix de categorii (precum „pisică" sau „câine"), este o problemă de clasificare. Unele feature-uri sunt intrări (ceea ce avem), iar altele sunt ieșiri (ceea ce vrem să prezicem).

În această teză, ne concentrăm pe două probleme de viziune computerizată: segmentarea semantică și estimarea adâncimii. Ambele sunt task-uri de predicție densă, ceea ce înseamnă că trebuie să facem o predicție pentru fiecare pixel dintr-o imagine.

Segmentare Semantică Această task implică atribuirea unei labeluri de clasă fiecărui pixel, cum ar fi identificarea tuturor pixelilor care aparțin clădirilor, drumurilor sau copacilor. Este o parte cheie a înțelegerii scenei. O provocare este că setul de clase este de obicei fixat de creatorii setului de date.

Estimarea Adâncimii Aici, scopul este de a estima distanța de la cameră la fiecare pixel dintr-o imagine. Această informație 3D este foarte utilă pentru robotică și navigație.

Rețele Neurale

Rețelele neurale reprezintă tehnologia principală utilizată în Învățarea Automată astăzi, precum și în această teză. Pentru a utiliza una, aveți nevoie de un set de date etichetat, o arhitectură de model și un algoritm de antrenament. Putem considera o rețea neurală ca o funcție $y = f(x)$, unde x sunt datele de intrare, y este ieșirea modelului, iar funcția f conține ponderi învățabile, W, care sunt ajustate în timpul antrenamentului.

O rețea poate fi *discriminativă*, adică prezice o ieșire (cum ar fi o etichetă de clasă sau o valoare). Sau poate fi *generativă*, adică creează date noi similare cu cele de intrare, cum ar fi generarea unei imagini din zgomot. Pentru probleme de clasificare (discriminative), este comun să se utilizeze un vector *one-hot* pentru a reprezenta clasele. De exemplu, cu trei clase, „pisică", „câine" și „rață", le-am reprezenta ca $[1, 0, 0]$, $[0, 1, 0]$ și $[0, 0, 1]$. În acest fel, modelul învață că toate predicțiile greșite sunt la fel de greșite.

Antrenarea Rețelelor Neurale

Antrenarea unei rețele neurale înseamnă ajustarea ponderilor sale W pe baza unui set de date. Utilizăm un algoritm numit retropropagare. Procesul este:

1. Trecem o intrare X_i prin model pentru a obține o predicție Y_i.
2. Calculăm eroarea (sau pierderea) dintre predicția Y_i și eticheta reală GT_i.
3. Actualizăm ponderile W pentru a reduce această eroare.

Pasul de actualizare se face de obicei folosind *gradient descent*, unde ajustăm ponderile în direcția care scade cel mai rapid eroarea. Scopul nu este doar de a memora datele de antrenament (*over-fitting*), ci de a învăța tipare care se *generalizează* la date noi, nevăzute. Pentru a asigura acest lucru, împărțim datele noastre în seturi de antrenament, validare și test. Modelul este antrenat pe setul de antrenament, performanța sa este monitorizată pe setul de validare pentru a preveni over-fittingul, iar evaluarea finală se face pe setul de test.

Rețele Neurale Profunde și diverse arhitecturi de modele

O rețea neurală adâncă este pur și simplu o rețea neurală cu mai multe straturi. Fiecare strat aplică o transformare ieșirii stratului anterior. Această structură stratificată permite rețelei să învețe o ierarhie de feature-uri. Straturile timpurii pot învăța tipare simple precum muchii și colțuri, în timp ce straturile mai adânci le combină pentru a recunoaște obiecte mai complexe. Figura 2.2 arată cum fiecare strat dintr-o rețea transformă datele, făcând în cele din urmă o problemă complexă, non-liniară, ușor de rezolvat pentru stratul final.

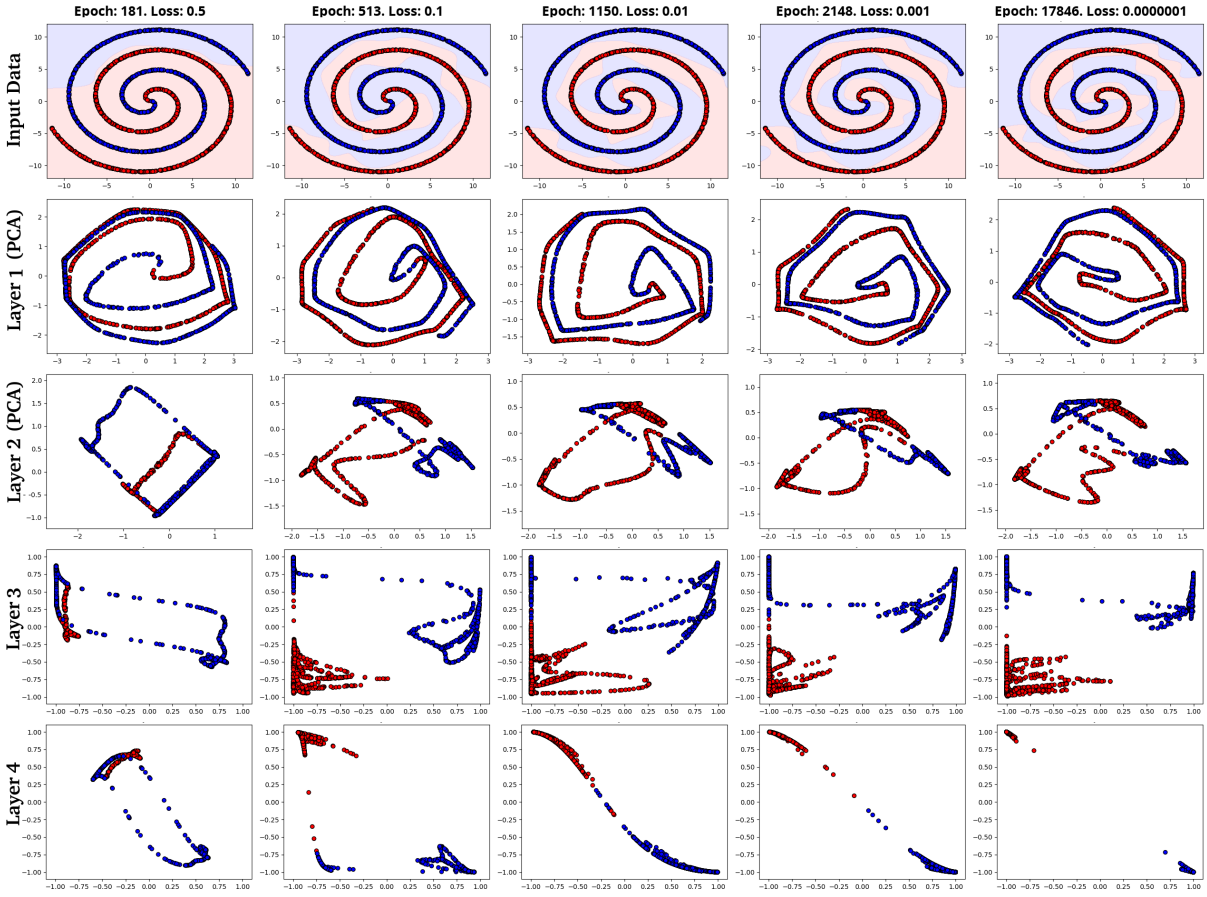


Figure 2.2: O rețea neurală cu patru straturi care învață să clasifice un set de date spiralat. Putem observa cum datele sunt transformate la fiecare strat, devenind liniar separabile la ultimul strat.

Rețele Neurale Convoluționale Pentru datele de imagine, un tip special de rețea adâncă numită Rețea Neuronală Convoluțională (CNN) este foarte eficient. Utilizăm acest tip de rețea neurală pe parcursul întregii teze. În loc să conecteze fiecare pixel de intrare la fiecare neuron din primul strat, o rețea CNN folosește filtre (sau kerneluri) mici, partajate, care glisează peste imagine, reducând numărul de operații. Aceste filtre sunt învățate în timpul antrenamentului și sunt bune la detectarea tiparelor locale precum muchii, texturi și forme, indiferent de locul în care apar în imagine. Aceste biasuri inductive ale imaginilor naturale fac rețelele CNN mult mai eficiente, necesitând mai puține date pentru a antrena distribuții de date complexe.

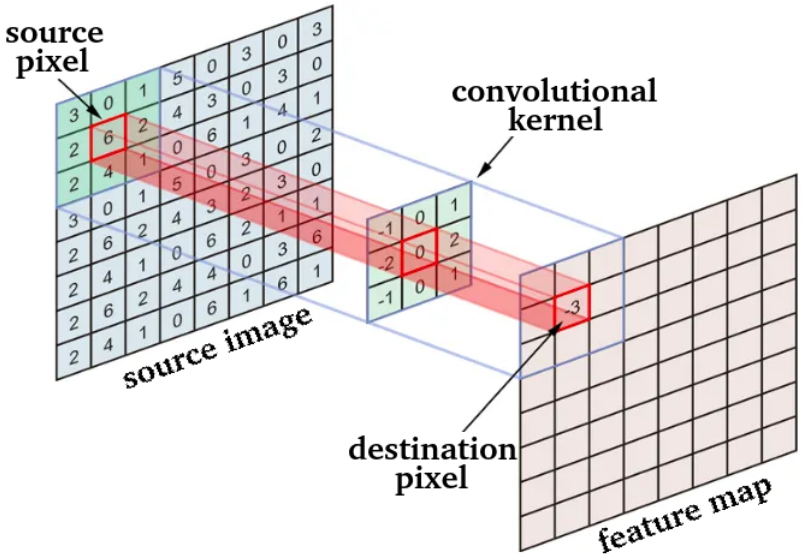


Figure 2.3: Operatorul convoluțional. Un kernel (filtru) mic glisează peste imaginea de intrare pentru a produce o hartă de feature-uri.

SafeUAV: Arhitectura Rețelei Neurale Convoluționale

Una dintre contribuțiile noastre timpurii a fost o rețea CNN integrabilă (embeddable) pentru estimarea adâncimii și identificarea zonelor sigure de aterizare (segmentare HVO). Am proiectat o rețea bazată pe populara arhitectură U-Net. Am creat două versiuni: *SafeUAV-Net-Large* pentru o acuratețe mai mare și *SafeUAV-Net-Small* pentru performanță în timp real pe hardware integrat (embedded) precum NVIDIA Jetson TX2.

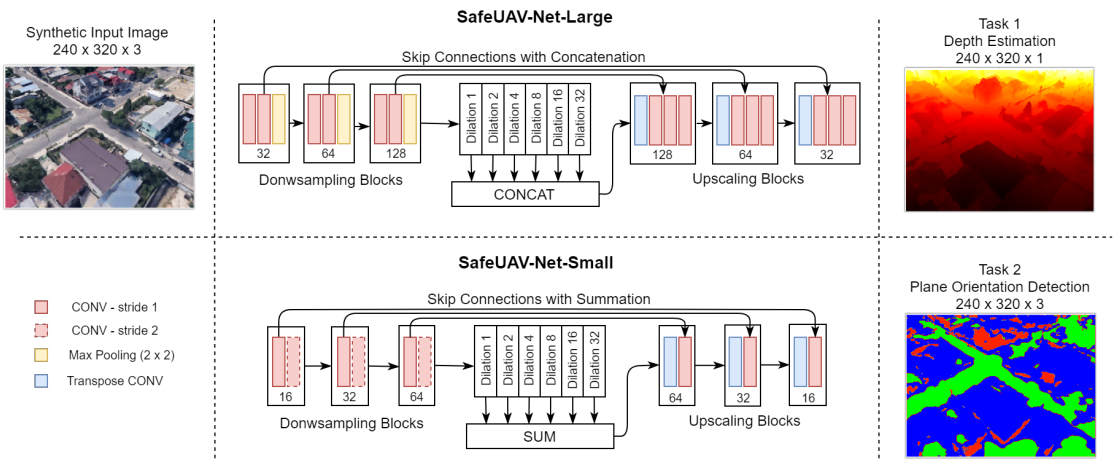


Figure 2.4: Arhitecturile de rețea SafeUAV propuse, utilizate pentru task-uri precum estimarea adâncimii și segmentarea semantică.

Ambele rețele folosesc o structură de tip codificator-decodificator cu conexiuni de tip „skip", ceea ce este tipic pentru U-Nets. Versiunea mai mică folosește mai puține filtre și un design de tip „bottleneck" mai eficient pentru a reduce costul computațional. Tabelul 2.2 prezintă o comparație a dimensiunii și vitezei lor.

Rețea	Număr de Parametri	Utilizare Memorie	FPS (Jetson TX2)
U-net [26]	31.031.745	1.7GB	37
DeepLabv3+ [5]	53.549.729	1.9GB	n/a
SafeUAV-Net-Large	23.896.129	927MB	35
SafeUAV-Net-Small	1.029.537	433MB	138

Table 2.2: Statistici de inferență pentru rețelele SafeUAV comparativ cu modelele standard.

Experimentele noastre au arătat că aceste arhitecturi au avut performanțe competitive, și chiar mai bune decât unele modele standard de la acea vreme, atât la estimarea adâncimii, cât și la segmentarea HVO pe setul nostru de date SafeUAV.

Învățare Multi-Modală și Multi-Task

În această teză, abordăm problema învățării multi-modale și multi-task. Într-un cadru Multi-Modal Multi-Task (MTL), avem mai multe tipuri de intrări (*modalități*) și/sau mai multe ieșiri de predicție (*task-uri*). De exemplu, am putea folosi atât imagini RGB, cât și date de adâncime ca intrare pentru a prezice atât segmentarea semantică, cât și localizarea obiectelor ca ieșire.

O provocare cheie în MTL este de a decide cum să combinăm intrările și cum să structurăm ieșirile. Putem fuziona intrările devreme (de ex., prin stivuirea canalelor de imagine) sau târziu (procesând fiecare intrare cu o rețea separată înainte de a le combina). Pentru ieșiri, putem folosi un decodificator partajat pentru toate task-urile sau decodificatoare separate, specializate pentru fiecare. Aceste alegeri depind de problema specifică. O problemă care poate apărea este *transferul negativ* (negative transfer), unde antrenarea pe un task afectează negativ performanța pe alta.

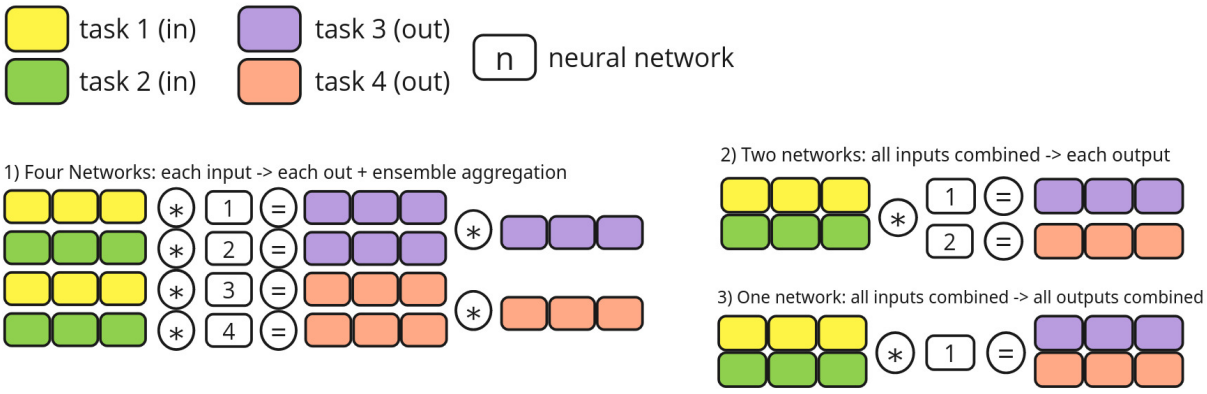


Figure 2.5: Diferite moduri de a modela o problemă cu 2 intrări și 2 ieșiri. Stânga: Modele separate. Dreapta sus: Intrări fuzionate, ieșiri separate. Dreapta jos: Intrări fuzionate, rețea de ieșire partajată.

Învățare prin Ansambluri

O rețea neurală standard oferă o predicție, dar nu ne spune cât de încrezătoare este. Învățarea prin Ansambluri (Ensemble Learning) este o tehnica de a aborda această problemă. Ideea este de a antrena mai multe modele independente pe aceeași task și apoi de a combina predicțiile lor. Dacă modelele sunt diverse și fac tipuri diferite de erori, predicția combinată este adesea mai precisă și mai fiabilă decât predicția oricărui model individual.

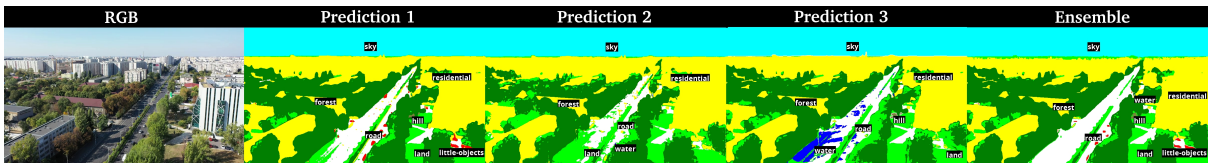


Figure 2.6: Un exemplu de Învățare prin Ansambluri. Trei modele diferite fac predicții cu erori diferite. Prin medierea lor, obținem o predicție finală care este mai curată și mai precisă.

Predicțiile sunt combinate folosind o funcție de agregare. Pentru task-urile de regresie, aceasta poate fi o medie simplă. Pentru clasificare, ar putea fi un vot majoritar. Cheia este ca modelele să fie diverse; dacă toate modelele fac aceleași greșeli, ansamblul nu va ajuta.

Grafuri

Un graf este o structură matematică formată din noduri (sau vârfuri) și muchii care le conectează. Grafurile sunt o modalitate puternică de a modela relațiile dintre lucruri. În Învățarea Automată, Rețelele Neurale pe Grafuri (GNN) au fost dezvoltate pentru a aplica conceptele de rețele neurale la date structurate sub formă de graf, cum ar fi rețelele sociale sau structurile moleculare.

În lucrarea noastră, folosim *grafuri de Reprezentări*. Aici, fiecare nod din graf reprezintă o întreagă perspectivă sau modalitate a datelor (cum ar fi imagini RGB sau hărți de adâncime), iar fiecare muchie este o rețea neurală care transformă o perspectivă în alta. Acest cadru ne permite să modelăm relațiile dintre diferite task-uri și modalități într-un mod structurat și oferă o punte naturală către Învățarea prin Ansambluri, deoarece mai multe căi către același nod creează un ansamblu de predicții. Poate fi văzut și ca un caz special de Modelare Grafică Probabilistică [17, 3].

Învățare Nesupervizată și Auto-supervizată

Ce se întâmplă dacă avem multe date, dar fără labeluri? Aici intervine învățarea nesupervizată. Acești algoritmi încearcă să găsească tipare și structuri în date pe cont propriu, de exemplu prin gruparea punctelor de date similare.

Un subset puternic al acesteia este *învățarea auto-supervizată*, unde modelul își creează propria supervizare din date. O tehnică comună este autoencoderul, care învață să comprime datele într-o reprezentare mai mică (codificare) și apoi să reconstruiască datele originale din aceasta (decodificare). Învățând să reconstruiască intrarea, modelul învață feature-uri semnificative despre date fără a avea nevoie de labeluri externe.

O altă metodă populară este Autoencoderul Mascat (Maked Auto Encoder, MAE). Într-un MAE, părți ale intrării (cum ar fi patch-uri dintr-o imagine sau cuvinte dintr-o propoziție) sunt ascunse aleatoriu, iar sarcina modelului este de a prezice părțile lipsă pe baza contextului vizibil. Acest lucru forțează modelul să învețe relații contextuale profunde în cadrul datelor.

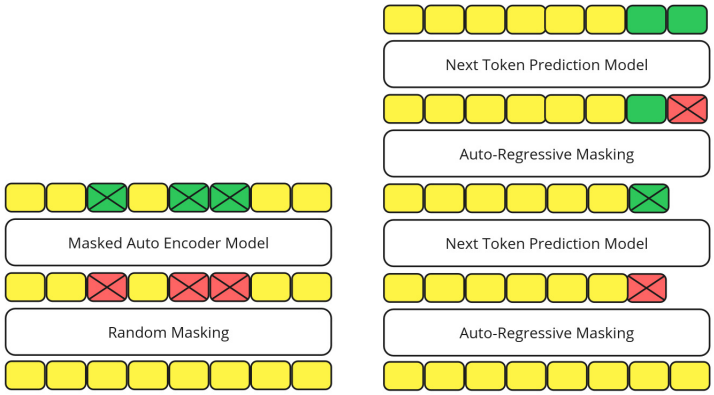


Figure 2.7: Două metode de învățare auto-supervizată. Autoencoderele Mascate (stânga) reconstruiesc părțile lipsă ale intrării. Predicția Următorului Element (dreapta) prezice următorul element dintr-o secvență.

Distilarea Modelelor

Distilarea Modelelor este o tehnică prin care cunoștințele de la un model mare și complex (*profesor*) sunt transferate la un model mai mic și mai eficient (*student*). Mai întâi, modelul profesor este antrenat pe un task. Apoi, modelul student este antrenat nu pe labelurile originale ale datelor, ci pe predicțiile modelului profesor.

Acest proces poate fi folosit pentru a comprima modele mari în unele mai mici care pot rula pe dispozitive cu resurse limitate. De asemenea, uneori poate duce la un model student care are performanțe mai bune decât dacă ar fi fost antrenat de la zero pe labelurile originale, deoarece predicțiile „soft" ale profesorului (probabilități în loc de labeluri "hard" one-hot) pot oferi un semnal de antrenament mai bogat.

Învățare Semi-supervizată

Învățarea semi-supervizată este o cale de mijloc între învățarea supervizată și cea nesupervizată. Se folosește atunci când aveți o cantitate mică de date labeluite și o cantitate mare de date nelabeluite. Procesul tipic este:

1. Antrenați un *model de pornire* (seed model) pe setul de date etichetat mic.
2. Folosiți acest model de pornire pentru a face predicții pe setul de date neetichetat mare. Aceste predicții se numesc *pseudo-labeluri*.
3. Combinați datele labeluite originale cu datele nou pseudo-labeluite.
4. Antrenați un model final, adesea mai mare, pe acest set de date combinat.

Această abordare poate fi foarte eficientă, mai ales atunci când pseudo-labelurile sunt generate

de un model fiabil, cum ar fi un ansamblu. Ne permite să valorificăm cantități vaste de date nelabeluite pentru a construi modele mai bune decât am putea doar cu datele labeluite. Aceasta este o tehnică cheie pe care o folosim în capitolele următoare.

Chapter 3

Învățarea prin ansamblu și distilare a metodelor neurale și analitice pentru estimarea adâncimii

Acest capitol prezintă lucrarea noastră privind estimarea adâncimii metrice nesupervizate pentru UAV-uri, care a fost prezentată pentru prima dată în articolul nostru *Depth Distillation: Unsupervised Metric Depth Estimation for UAVs by Finding Consensus Between Kinematics, Optical Flow and Deep Learning* [23].

Estimarea adâncimii precise, din lumea reală (metrică), este importantă pentru ca UAV-urile să navigheze în siguranță. Realizarea acestui lucru fără supervizare directă sau date de odometrie este o problemă dificilă. Pe de altă parte, putem calcula adâncimea folosind matematica mișcării camerei (cinematică) și fluxul optic. Această metodă analitică este exactă în teorie, dar poate fi instabilă și eșuează complet în unele zone, cum ar fi focusul de expansiune. Propunem un model care combină ce e mai bun din ambele lumi: precizia metodelor analitice și robustețea învățării adânci nesupervizate.

Principalele noastre contribuții sunt:

1. O nouă metodă pentru estimarea adâncimii metrice pentru UAV-uri folosind doar o cameră RGB. Aceasta învață într-un mod nesupervizat dintr-un singur videoclip de zbor, folosind un ansamblu format dintr-o metodă analitică și o metodă de învățare adâncă ca „profesor" pentru a distila cunoștințele într-o rețea finală.
2. O metodă analitică îmbunătățită pentru estimarea adâncimii, care este mai robustă deoarece estimează simultan adâncimea și corectează erorile din viteza unghiulară a camerei.
3. Un nou set de date UAV cu aproape 20 de minute de video de zbor, incluzând cinematica vehiculului și GPS. Acest set de date este extins mai târziu în această teză, creând setul de date Dronescapes.

După cum se arată în Figura 3.1, folosim două căi pentru a estima adâncimea. Calea *analitică* utilizează odometria și fluxul optic. Calea *bazată pe date* folosește o rețea adâncă nesuper-

vizată. Aceste două căi supervizează o singură rețea adâncă finală. Apoi, prin distilarea modelului, rețeaua finală învață să estimeze adâncimea metrică pe baza unui consens între geometrie, mișcarea camerei și imaginea de intrare. Rețeaua rezultată este mică și rapidă, ceea ce o face perfectă pentru utilizarea pe dispozitive integrate (embedded). Pentru a obține o adâncime metrică precisă, folosim mai multe metode. O rețea nesupervizată furnizează adâncime non-metrică (D_{Unsup}), în timp ce o altă metodă folosește odometria și fluxul optic pentru a obține adâncime metrică ($D_{OdoFlow}$). Folosim $D_{OdoFlow}$ pentru a oferi o scară reală lui D_{Unsup}. Împreună, acestea două formează un ansamblu „profesor" care antrenează o rețea student finală. Doar pentru evaluare, folosim o conductă (pipeline) de Structură din Mișcare (SfM) pentru a genera o a treia hartă de adâncime, D_{SfM}, care servește drept referință (ground truth). Deși o conductă SfM este lentă, poate produce hărți de adâncime precise.

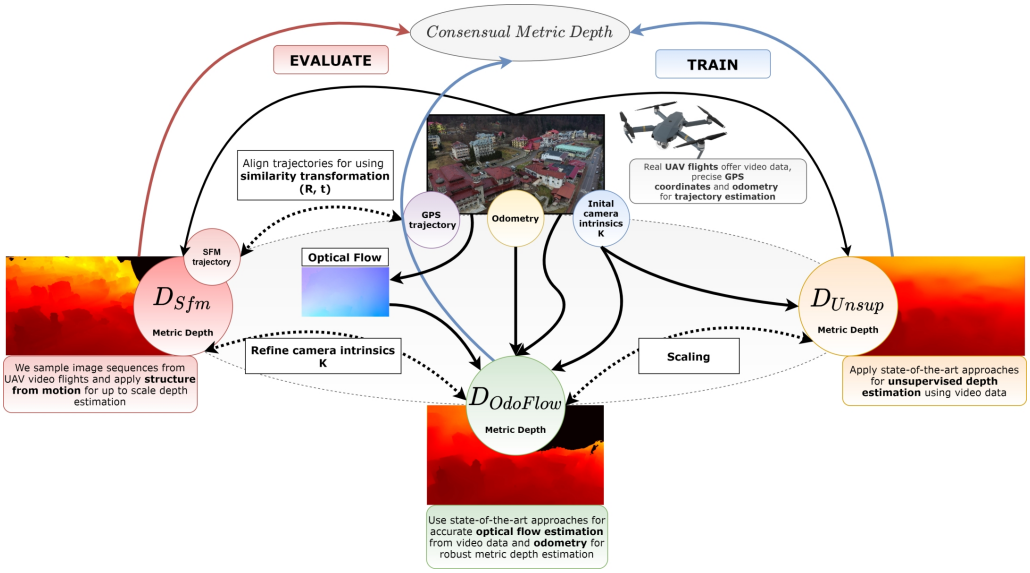


Figure 3.1: Prezentare generală a metodei noastre, combinând căi analitice și bazate pe date.

3.1 Prezentare generală a distilării adâncimii metrice

Estimarea traiectoriei din GPS Netezim datele GPS zgomotoase prin potrivirea polinoamelor de gradul 3 peste o fereastră glisantă de măsurători. Acest lucru ne oferă o estimare mai stabilă a traiectoriei.

Adâncime analitică din odometrie, flux și traiectorie Relația dintre mișcarea camerei, fluxul optic și adâncimea metrică este descrisă de *Matricea Jacobiană a Imaginii*. Dacă cunoaștem vitezele liniare și unghiulare ale camerei și putem calcula fluxul optic între cadre, putem rezolva pentru o hartă de adâncime metrică densă. Introducem, de asemenea, o modalitate mai robustă de a face acest lucru, mai puțin sensibilă la erorile de flux optic.

Mișcarea unui pixel este legată de viteza liniară a camerei ν și viteza unghiulară ω prin matricea Jacobiană a Imaginii, așa cum se arată în ecuația de mai jos. Din această relație, putem forma o ecuație liniară în termeni de adâncime inversă $1/Z$: $(J_\nu \nu)\frac{1}{Z} + J_\omega \Delta\omega = \dot{p} - J_\omega \omega$, unde $\dot{\tilde{p}}$ este fluxul optic. Soluția celor mai mici pătrate pentru adâncimea Z este atunci: $Z = \frac{\|A\|^2}{A^T b}$

Cu toate acestea, această soluție poate fi sensibilă la erorile din viteza unghiulară estimată. Pentru a o face mai robustă, introducem un termen de corecție $\Delta\omega$ și rezolvăm simultan atât pentru adâncimile Z_i, cât și pentru corecție, peste mai multe puncte. Aceasta este una dintre contribuțiile noastre teoretice.

Post-procesare pentru calculul adâncimii Soluția pentru Z din ecuația de mai sus devine instabilă în apropierea focusului de expansiune, unde fluxul optic este aproape de zero. Filtrăm aceste valori de adâncime nesigure prin aplicarea unor praguri asupra magnitudinii fluxului optic și a unghiului dintre vectorii de mișcare.

Aliniere SfM Pentru evaluare, folosim un instrument SfM offline pentru a reconstrui un model 3D al scenei. Deoarece acest model nu este la o scară reală, îi aliniem traiectoria cu traiectoria noastră GPS pentru a o face metrică. Acest lucru ne oferă o hartă de adâncime de înaltă calitate (D_{SfM}) pe care să o folosim ca referință pentru experimentele noastre.

Sistemul nostru, prezentat în Figura 3.2, combină ieșirile metodei analitice ($D_{OdoFlow}$) și ale rețelei nesupervizate (D_{Unsup}) pentru a forma un ansamblu profesor. O rețea student este antrenată să imite acest profesor, învățând să prezică adâncimea metrică dintr-o singură imagine RGB. Predicțiile studentului sunt evaluate față de referința offline D_{SfM}, folosind o pierdere L1 care ignoră pixelii invalizi din harta SfM.

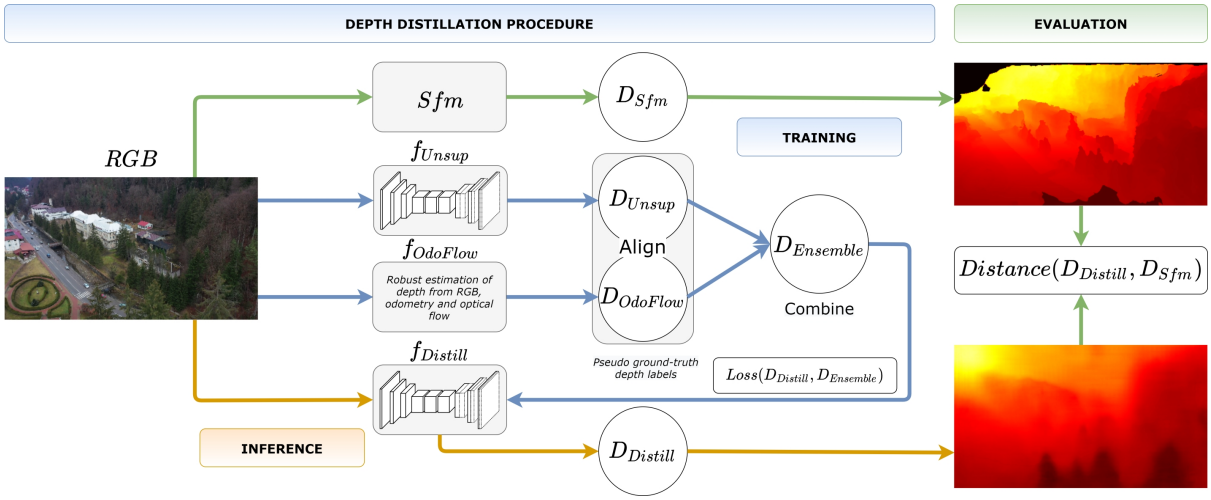


Figure 3.2: Procesul de distilare. Metodele analitică ($D_{OdoFlow}$) și nesupervizată (D_{Unsup}) sunt combinate într-un ansamblu profesor, care antrenează o rețea student pentru a prezice adâncimea metrică dintr-o singură imagine RGB.

3.2 Analiză Experimentală

Introducem un nou set de date din două zboruri UAV deasupra unor localități montane, pe care le numim **Slanic** și **Herculane** (vezi traiectoriile în Figura 3.3). Slanic este împărțit în seturi de antrenament și testare pentru a evalua performanța într-o scenă familiară. Herculane

este folosit doar pentru testare pentru a vedea cât de bine se generalizează metoda noastră la un mediu nou.

Figure 3.3: Traiectoriile de zbor ale UAV-urilor pentru seturile de date Slanic și Herculane.

Configurare și Rezultate

Folosim instrumentul Meshroom [13] pentru reconstrucția SfM, RAFT [29] pentru fluxul optic și DPT [25] pentru adâncimea nesupervizată. Rețelele student sunt două variante ale arhitecturii SafeUAV, Tiny și Large, care sunt proiectate pentru performanță în timp real pe sisteme integrate. Antrenăm studentul folosind o pierdere L2 față de ansamblul profesor, care este creat prin medierea la nivel de pixel a $D_{OdoFlow}$ și a D_{Unsup} scalat.

Rezultatele noastre arată că rețeaua student distilată poate depăși performanța profesorilor săi. Așa cum se arată în Tabelul 3.1, pe setul de testare Slanic, rețeaua student obține o eroare mai mică (21.58 m) decât metoda nesupervizată (27.28 m), metoda analitică (26.05 m) și chiar ansamblul profesor însuși (25.63 m). Aceeași tendință se menține și atunci când ne uităm doar la zonele „bune" unde metoda analitică este validă.

	Slanic		Herculane	
	Metrică	Relativă	Metrică	Relativă
D_{Unsup}	27.28 m	17.10 %	44.39 m	20.29 %
$D_{OdoFlow}$	26.05 m	16.34 %	**39.67 m**	**17.53 %**
$D_{Ensemble}$	25.63 m	15.88 %	41.18 m	18.29 %
$Tiny - 16$	**21.58 m**	**14.58 %**	46.77 m	24.09 %
$Large - 16$	21.84 m	14.65%	48.00 m	23.97 %

Table 3.1: Erorile absolute medii și relative față de datele de referință D_{SfM}. Modelele student distilate depășesc performanța profesorilor lor pe setul de testare Slanic.

Am constatat, de asemenea, că erorile sunt mult mai mici pentru obiectele care sunt mai aproape de UAV, care este regiunea cea mai importantă pentru task-uri precum evitarea obstacolelor. Rezultatele calitative din Figura 3.4 arată că rețeaua student învață să combine punctele forte ale ambelor metode profesor, producând hărți de adâncime netede și detaliate.

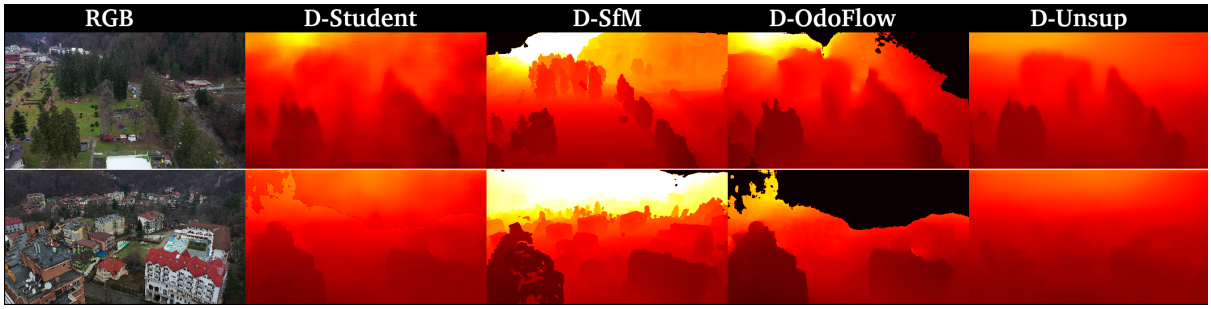


Figure 3.4: Rezultate calitative. De la stânga: RGB, Student, SfM, OdoFlow, Nesupervizat. Rețeaua student produce hărți de adâncime dense și precise, învățând să corecteze erorile profesorilor săi.

În final, rețelele noastre student sunt foarte eficiente. În special, modelul Tiny-16 rulează la peste 10 FPS pe o platformă integrată Jetson TX2, ceea ce îl face potrivit pentru implementare în timp real pe un UAV.

3.3 Concluzii

Nicio metodă unică pentru estimarea adâncimii metrice nu este perfectă. Cu toate acestea, prin combinarea unei metode analitice cu o abordare de învățare adâncă nesupervizată, putem crea un ansamblu profesor puternic. Acest profesor își poate distila apoi cunoștințele într-o singură rețea student ușoară, care este rapidă, robustă și adesea mai precisă decât profesorii săi. Evaluarea noastră pe un set de date UAV nou și provocator arată că acest antrenament auto-supervizat este eficient și practic pentru aplicații din lumea reală.

Chapter 4

Consens pe Grafuri Neurale Multi-Strat pentru Învățare Semi-supervizată

Acest capitol rezumă lucrarea din articolul nostru, *Semi-Supervised Learning for Multi-Task Scene Understanding by Neural Graph Consensus* [18], care a fost acceptat la *a 35-a Conferință AAAI despre Inteligență Artificială (AAAI 2021)*.

Abordăm problema învățării semi-supervizate atunci când avem mai multe moduri diferite de a interpreta o scenă vizuală, în special pentru înțelegerea imaginilor aeriene. Metoda noastră folosește un graf de rețele neurale pentru a găsi un consens între aceste interpretări. În acest graf, fiecare nod reprezintă un strat de interpretare (cum ar fi adâncimea sau segmentarea semantică), iar fiecare muchie este o rețea adâncă care transformă un strat în altul.

Procesul de învățare are două faze:

- faza supervizată: antrenăm rețelele de muchie folosind datele de referință (ground truth) disponibile.
- faze semi-supervizate (≥ 1): folosim date nelabeluite generate din faza anterioară, supervizate sau semi-supervizate pe date nou adăugate.

Pseudo-labelurile sunt generate prin găsirea unui consens între mai multe căi din graf care duc toate la aceeași ieșire. Acestea acționează ca ansambluri profesor, iar pseudo-labelurile sunt folosite pentru a antrena un model student pentru faza următoare. Prin acest proces iterativ, întregul graf devine mai consistent, chiar și fără labeluri noi.

Această metodă se numește **Consens pe Grafuri Neurale (NGC)** și reunește punctele forte ale rețelelor adânci și ale grafurilor. Rețelele adânci sunt puternice, dar necesită o cantitate mare de date labeluite. Grafurile pot găsi soluții globale din informații locale. Așa cum se arată în Figura 4.1, NGC poate conecta multe task-uri diferite, cum ar fi prezicerea structurii 3D, a pozei și a claselor semantice, într-un singur cadru. Forțând aceste task-uri să fie de acord între ele, ele pot învăța eficient din date nelabeluite. Contribuția noastră principală este modelul NGC, un nou cadru pentru învățarea semi-supervizată a interpretărilor multiple ale scenei. Arătăm cum diferite task-uri se pot învăța reciproc prin consens, oferim suport

teoretic pentru abordarea noastră și demonstrăm eficacitatea acesteia pe un set de date la scară largă.

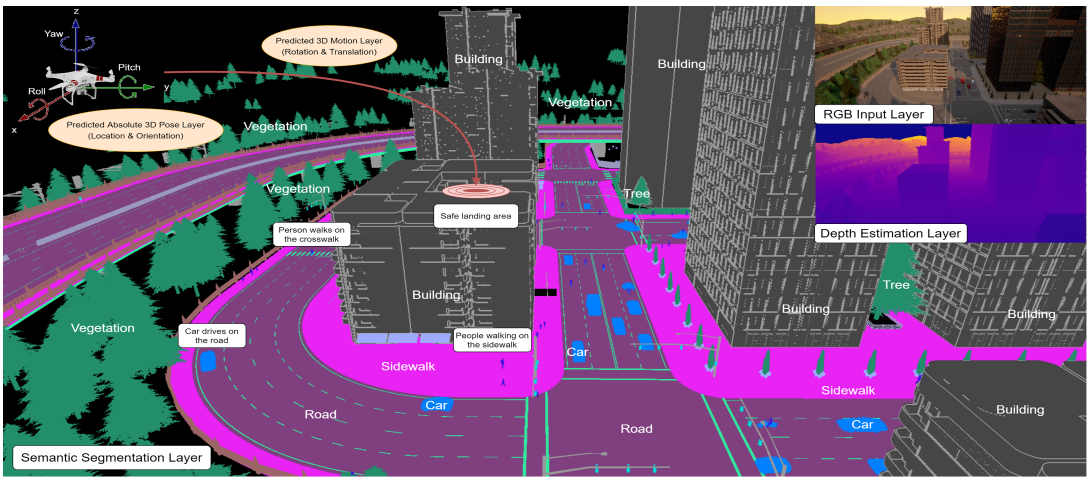
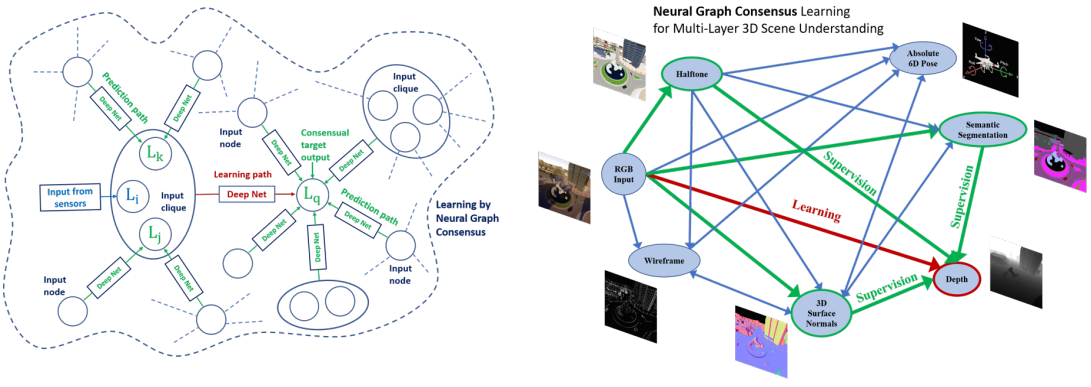


Figure 4.1: Diferite interpretări ale scenei, precum structura 3D, poza și semantica, sunt conectate într-un graf neuronal. Căile multiple care ajung la același nod pot acționa ca un "profesor" prin consens pentru a antrena o singură rețea de muchie. Acest lucru permite modelului să învețe robust din date nelabeluite.

4.1 NGC: Modelul de Consens pe Grafuri Neurale

În modelul NGC, fiecare nod i conține un strat L_i, care este o interpretare specifică a lumii (de ex., o hartă de adâncime sau o segmentare semantică). Muchiile din graf sunt rețele adânci care prezic un strat la un nod din straturile altor noduri. Acest model este prezentat în Figura 4.2.



A. General Conceptual NGC Model

B. Actual NGC model implemented

Figure 4.2: Modelul NGC: generic și implementat.

În stânga, modelul teoretic al grafului poate avea conexiuni complexe, unde multe căi pot

ajunge la un nod dat. Modelul funcționează prin alternarea rolurilor între rețelele de muchie. O rețea devine „studentul" și este antrenată folosind pseudo-datele de referință generate de consensul tuturor celorlalte căi care ajung la același nod de ieșire. Aceste alte căi acționează ca „profesori". În dreapta, prezentăm structura specifică pe care am folosit-o în experimentele noastre pentru a învăța adâncimea, segmentarea semantică și poziția unei drone într-un mediu simulat.

Analiză Teoretică

Procesul iterativ de învățare semi-supervizată al NGC constă în trei pași:

1. Pre-antrenarea unui set de rețele-muchie de pornire pe datele labeluite disponibile.
2. Formarea grafului NGC prin conectarea muchiilor individuale.
3. Pe un nou set neetichetat, re-antrenarea rețelelor-muchie folosind ieșirea consensuală a tuturor căilor care ajung la un nod specific ca pseudo-date de referință. Se repetă Pasul 3 iterativ până când nu mai sunt disponibile date noi.

Analiza noastră arată că pentru task-urile de regresie, această abordare de învățare prin ansamblu minimizează varianța dintre ieșirile diferitelor căi. Acest lucru duce la prima noastră propoziție:

Propoziția 1 Într-un graf NGC dens conectat, ne așteptăm ca varianța ieșirilor care ajung la un nod dat să scadă în timpul învățării prin ansamblu.

Validăm această propoziție experimental, demonstrând o variantă redusă, deci o consistență crescută a predicțiilor pentru task-urile de regresie. Pentru task-urile de clasificare, unde consensul se realizează prin vot, arătăm că acuratețea ansamblului profesor se îmbunătățește pe măsură ce adăugăm mai multe căi independente.

Propoziția 2 Dacă probabilitatea de succes p pentru o rețea de muchie este mai bună decât aleatorie, probabilitatea de succes a ansamblului profesor folosind votul majoritar tinde spre 1 pe măsură ce numărul de căi $N \to \infty$.

Simulările noastre susțin aceste constatări teoretice, indicând că performanța se îmbunătățește cu mai multe căi în ansamblu și poate continua să se îmbunătățească pe parcursul mai multor iterații. Această analiză face câteva presupuneri, cum ar fi faptul că fiecare rețea învață tipare complementare. Evident, dacă toți candidații ar produce același rezultat, varianța ar fi 0, dar și beneficiul adăugat de fiecare candidat ar fi nul. Corolarul este că avem nevoie de candidați diverși, ceea ce se poate obține având reprezentări multiple în seturile noastre de date, cum ar fi RGB (culori), muchii (texturi de nivel scăzut), adâncime (viziune de nivel mediu) și semantică (nivel înalt).

4.2 Analiză experimentală

Set de date și Implementare

Am creat un set de date la scară largă dintr-un mediu virtual folosind CARLA [8], unde o dronă zboară deasupra unui oraș, așa cum se vede în Figura 4.3. Pentru fiecare imagine, avem date de referință pentru adâncimea scenei, normalele suprafeței 3D, poza 6D, wireframes și segmentare semantică în 12 clase. Am implementat cadrul nostru NGC în PyTorch, folosind rețele ușoare de aproximativ 1.1M parametri pentru fiecare muchie. Graful complet este format din 27 de astfel de rețele.

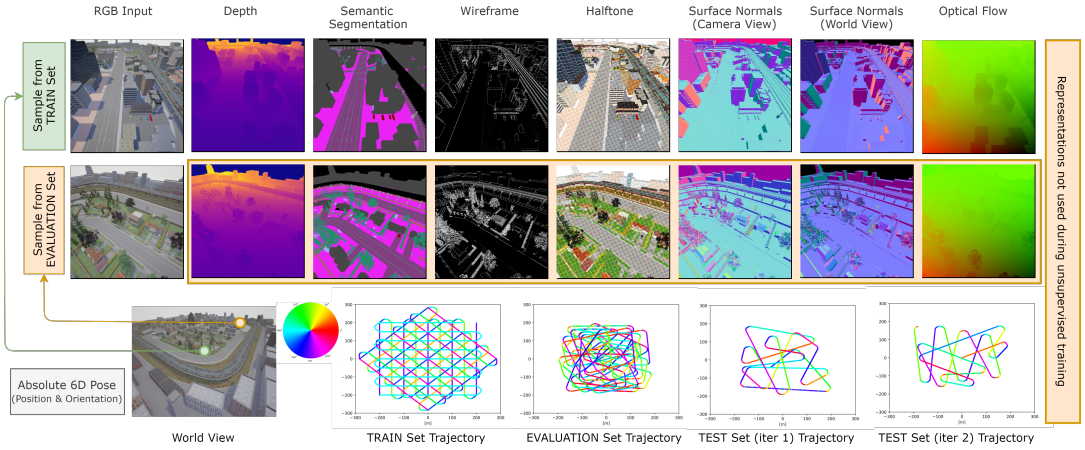


Figure 4.3: Eșantioane din setul nostru de date sintetic, arătând imagini RGB și traiectoriile dronei.

Învățare iterativă semi-supervizată prin ansamblu

Am urmat procesul de antrenament iterativ pentru trei iterații. Prima iterație a fost antrenament supervizat. Următoarele două au fost semi-supervizate, folosind date nelabeluite. Rezultatele, prezentate în Tabelul 4.1, arată o îmbunătățire constantă pentru toate task-urile și atât pentru rețelele unice „distilate", cât și pentru ansamblurile NGC complete. Acest lucru demonstrează că abordarea noastră bazată pe consens valorifică eficient datele nelabeluite pentru a îmbunătăți performanța. Rezultatele calitative din Figura 4.4 ilustrează în continuare aceste îmbunătățiri.

Reprezentare	Metrica de Evaluare	Iterația 1	Iterația 2		Iterația 3	
		EdgeNet	NGC	Distil. EdgeNet	NGC	Distil. EdgeNet
Adâncime	L1 (metri)	4.9844	3.4867	4.2802	**3.2994**	**3.9508**
	Pixeli ↑ (%)	-	79.30	60.66	**79.69**	**61.90**
Semantic Segmentare	mIOU	0.4840	0.4978	0.4980	**0.5258**	**0.5159**
	Pixeli ↑ (%)	-	79.46	69.62	**81.49**	**71.95**
Poziție	L2 (metri)	25.7597	15.5383	20.0204	**12.0764**	**15.5599**
Orientare	L1 (grade)	3.8439	2.5001	3.3961	**2.2088**	**3.0005**

Table 4.1: Rezultate de-a lungul a două iterații de învățare nesupervizată, arătând o îmbunătățire constantă atât pentru ansamblurile NGC (**roșu**), cât și pentru rețelele EdgeNet unice distilate (**albastru**). (Tabel prescurtat pentru concizie).

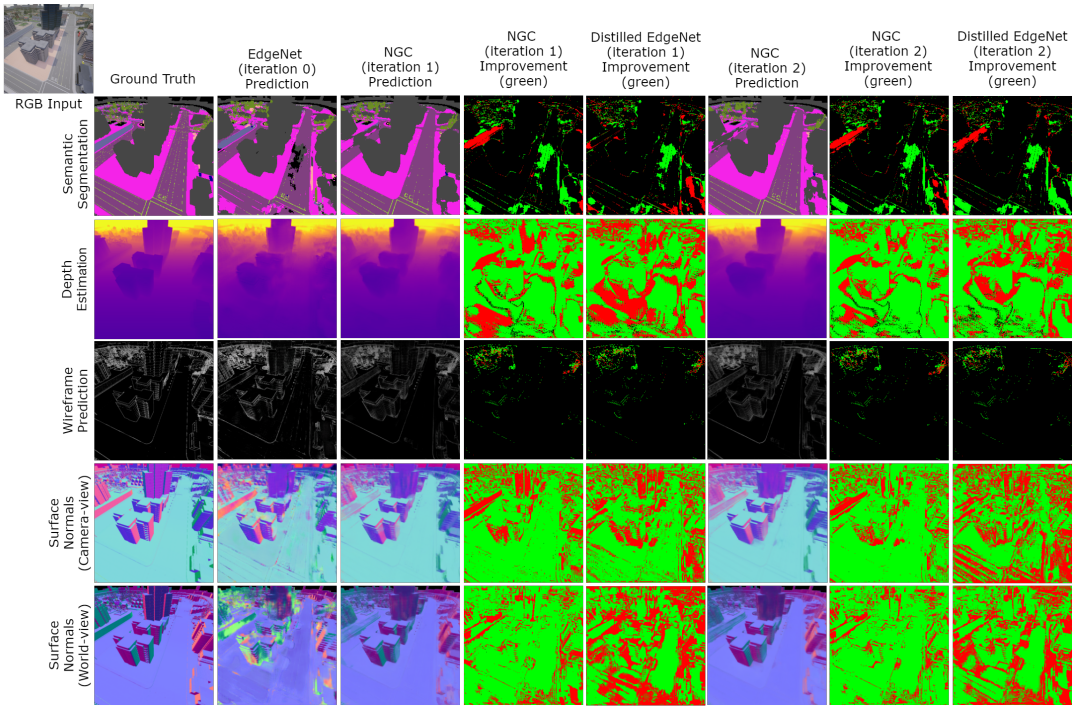


Figure 4.4: Rezultate calitative pentru NGC. Verdele arată zonele de îmbunătățire după învățarea semi-supervizată, în timp ce roșul arată zonele de degradare.

Comparații cu Stadiul Actual al Tehnologiei

Am comparat NGC cu metode de top din mai multe domenii conexe.

- **Învățare Multi-Task:** Am testat împotriva NDDR [10] și MTL-NAS [11]. Deși acestea au avut performanțe bune la un task (normale), metoda noastră a arătat o performanță generală mai bună, în special la segmentarea semantică.

- **Învățare Semi-supervizată:** Am comparat cu CCT [22], o metodă semi-supervizată generală. NGC-ul nostru a depășit semnificativ CCT la segmentarea semantică pe setul nostru de date, atât în scoruri absolute, cât și în îmbunătățirea obținută din date nelabeluite.
- **Metode de Ansamblu:** Am comparat NGC cu ansambluri standard de rețele. NGC, care utilizează reprezentări intermediare diverse, a avut performanțe mai bune decât ansamblurile de rețele antrenate toate pentru aceeași task.

4.3 Concluzii

Am introdus modelul de Consens pe Grafuri Neurale (NGC), o nouă metodă pentru învățarea semi-supervizată multi-task. NGC combină multe rețele adânci într-o singură structură de graf unde acestea învață una de la cealaltă prin consens reciproc. Experimentele noastre, susținute de analiza teoretică, arată că această abordare este foarte eficientă. Modelul învață cu succes șapte interpretări diferite ale scenei din imagini unice cu performanțe de top, demonstrând că învățarea din consens în grafuri mari, colaborative, este o direcție promițătoare pentru învățarea nesupervizată și semi-supervizată.

Chapter 5

Hiper-Grafuri Multi-Strat pentru Învățare Semi-Supervizată

Acest capitol rezumă munca noastră privind extinderea învățării bazate pe grafuri la hipergrafuri, așa cum este introdusă în publicațiile noastre [20, 24]. Construim pe capitolul anterior prin introducerea de hiper-muchii mai complexe și ansambluri învățabile și validăm metoda noastră pe două scenarii reale provocatoare: înțelegerea imaginilor aeriene și observarea Pământului. Graful poate conține acum hiper-noduri și hiper-muchii care constau în reprezentări multiple simultan (de ex., RGB plus muchii), alături de muchii regulate. Aplicăm procesul de învățare iterativă semi-supervizată introdus anterior. Acest proces permite sistemului să se îmbunătățească în timp, pe măsură ce mai multe date nelabeluite devin disponibile, chiar și atunci când adnotările sunt rare.

Introducem două seturi de date diferite: Dronescapes, un nou set de date video UAV, și setul de date NASA Earth Observations (NEO), care conține 22 de ani de date satelitare. Așa cum se arată în Figura 5.1, abordarea noastră se dovedește eficientă atât pentru înțelegerea scenelor urbane, cât și pentru observarea Pământului la scară largă, îmbunătățind acuratețea și coerența temporală.

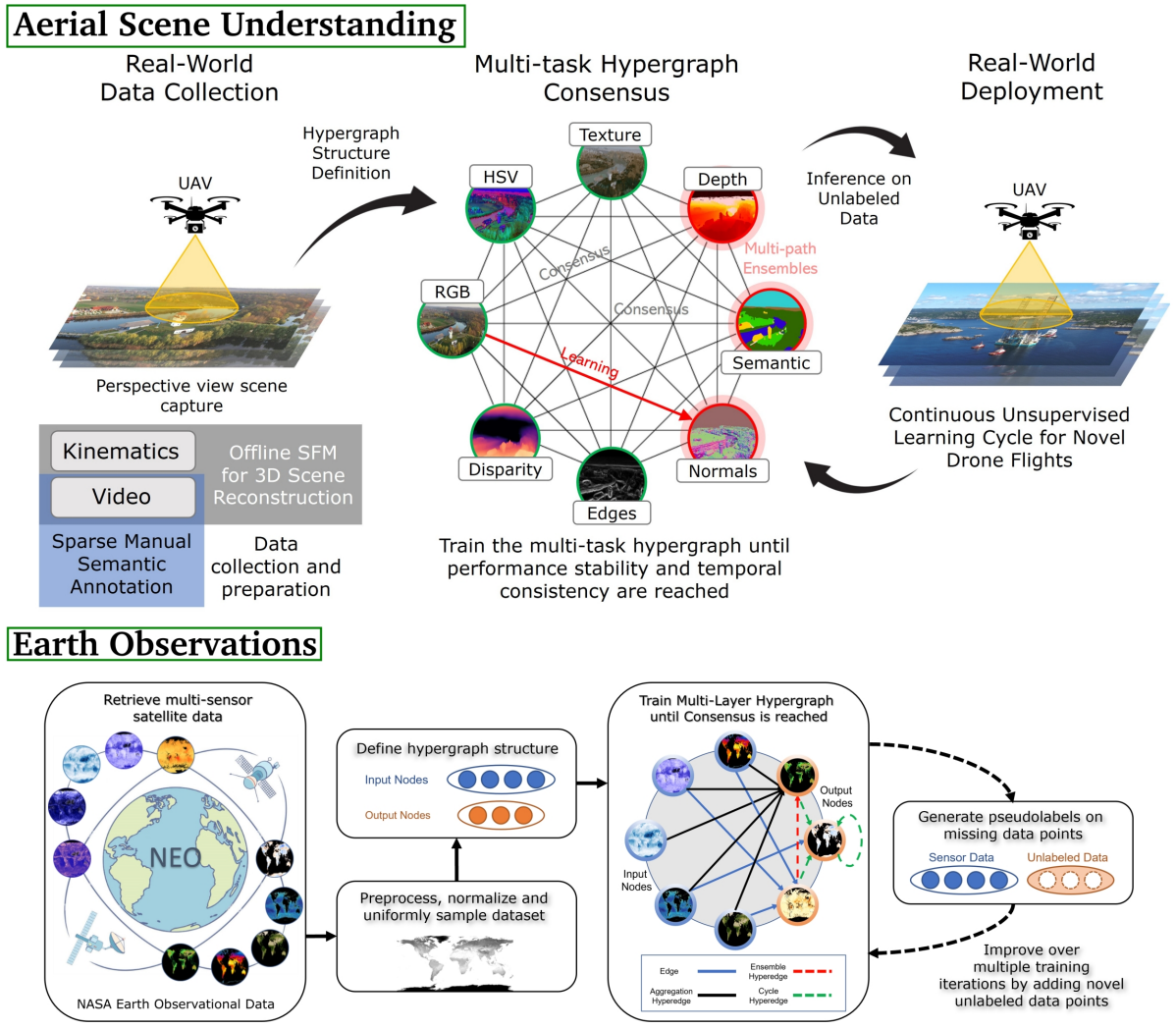


Figure 5.1: Hiper-Graful nostru multi-strat semi-supervizat în contextul UAV-urilor reale și al Observării Pământului. Definim structura Hiper-Grafului în termeni de noduri de intrare (de la senzori) și noduri de ieșire (cele care vor fi prezise). Antrenăm într-o manieră semi-supervizată de-a lungul mai multor iterații, ceea ce îmbunătățește atât acuratețea, cât și coerența temporală.

Lucrarea noastră se situează la intersecția mai multor domenii. În timp ce majoritatea metodelor de învățare nesupervizată se concentrează pe un număr limitat de task-uri, modelul nostru este general și poate gestiona multe task-uri simultan. Spre deosebire de metodele bazate pe consens care folosesc transformări simple ale imaginilor, noi ne concentrăm pe reprezentări semnificative ale scenei și învățăm să le echilibrăm. În domeniul învățării bazate pe grafuri, lucrările anterioare folosesc adesea structuri de graf mai simple, cu muchii pereche și ansambluri neînvățate. Abordarea noastră introduce hiper-muchii de ordin superior și învață mecanismul de ansamblu însuși. În final, folosim o formă de auto-distilare, unde „profesorii" puternici de ansamblu dintr-o iterație ghidează antrenamentul „studenților" muchii mai simple în următoarea, un concept care se bazează pe principiile consacrate ale distilării cunoștințelor.

5.1 Modelul de Hiper-Graf Multi-Strat

Structura Hiper-Grafului și Tipuri de Muchii

Hiper-Graful nostru Multi-Task, prezentat în Figura 5.2, constă în noduri de intrare (N_i), reprezentând date cunoscute de la senzori, și noduri de ieșire (N_o), reprezentând perspectivele asupra lumii pe care dorim să le prezicem. Aceste noduri sunt conectate prin diferite tipuri de muchii și hiper-muchii, care sunt modelate de rețele neurale U-Net mici.

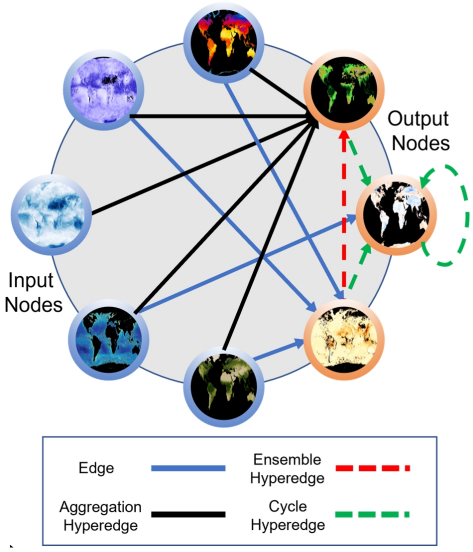


Figure 5.2: Hiper-Graf Multi-Task. Noduri de intrare și de ieșire, muchii și hiper-muchii.

Ca o extensie a capitolului anterior, care folosea doar muchii simple, introducem mai multe tipuri noi de hiper-muchii pentru a captura relații mai complexe între straturi, așa cum este ilustrat în Figura 5.3:

- **Muchii de un singur salt (E)**: O transformare simplă de la un nod de intrare la un nod de ieșire.
- **Muchii de două salturi (TH-E)**: O cale care trece printr-un nod prezis intermediar.
- **Hiper-Muchii de Ansamblu (EH)**: Combină toate predicțiile de un singur salt pentru un nod pentru a ajuta la prezicerea altor noduri.
- **Hiper-Muchii de Agregare (AH)**: Concatenează toate nodurile de intrare pentru a prezice un singur nod de ieșire.
- **Hiper-Muchii de Ciclu (CH)**: Folosește toate intrările și ieșirile hiper-muchiilor AH pentru a face o predicție finală pentru un nod.

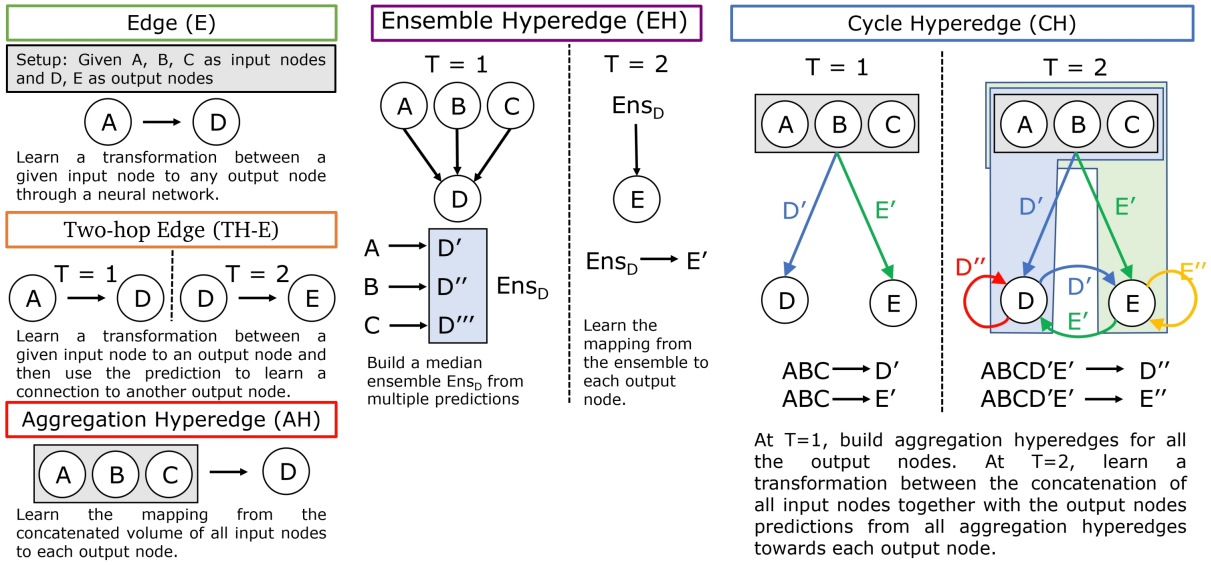


Figure 5.3: Tipuri de muchii și Hiper-Muchii în Hiper-Graf. În Capitolul 4, am folosit doar muchii (E și $TH-E$), în timp ce aici introducem hiper-muchii (AH, EH și CH) pentru a captura relații complexe între straturi.

Antrenament Iterativ Semi-Supervizat și Ansambluri

Reutilizăm același algoritm de antrenament iterativ semi-supervizat descris în capitolul anterior în Secțiunea 4.1. O contribuție nouă cheie este metoda noastră de **învățare a ansamblurilor de hiper-muchii**. În loc să mediem pur și simplu predicțiile candidaților, introducem mai multe modele învățabile, prezentate în Figura 5.4. Acestea variază de la un model liniar simplu (**S-L**$_{FW}$) la rețele neurale mai complexe care pot pondera dinamic fiecare predicție candidată la nivel de pixel (**S-NN**$_{DPW}$) sau pot învăța o mapare directă la ieșirea finală (**S-NN**$_{D}$). Aceste ansambluri învățate acționează ca profesori puternici, ghidând procesul de învățare semi-supervizată.

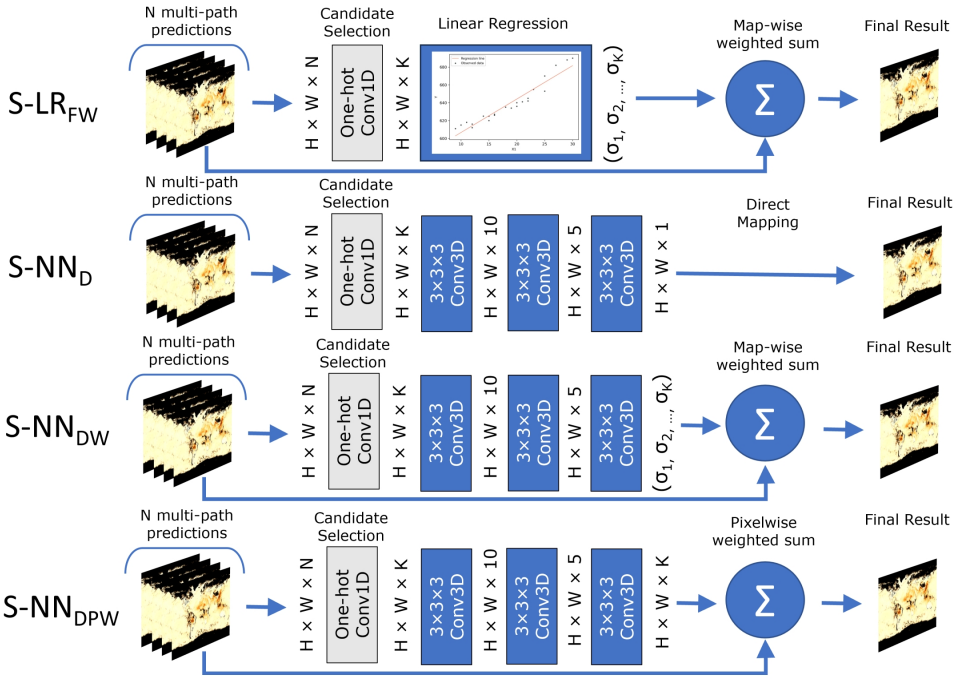


Figure 5.4: Arhitecturi de Ansamblu. Introducem patru tipuri de ansambluri, toate cu un model inițial învățabil de selecție a candidaților, care păstrează doar candidații relevanți înainte de a-i combina.

5.2 Analiză experimentală pe setul de date Dronescapes

Set de date și Configurare Experimentală

Introducem Dronescapes, un nou set de date video UAV la scară largă, cu scene diverse rurale, urbane și de coastă, așa cum se arată în Figura 5.5. Acesta include odometrie, informații 3D și adnotări manuale rare pentru segmentare semantică, făcându-l un punct de referință provocator din lumea reală.

Pentru experimentele noastre, am configurat o procedură de învățare iterativă în care începem cu un set mic de cadre labeluite manual. În iterațiile ulterioare, folosim hiper-graful nostru pentru a genera pseudo-labeluri pentru cadre video noi, nelabeluite, și reantrenăm modelul pe acest set de date extins. Ne concentrăm pe prezicerea a trei task-uri: segmentare semantică, adâncime și normalele suprafeței.

Figure 5.5: Cadre eșantion din fiecare dintre cele 10 scene din setul de date Dronescapes, arătând o mare variație în peisaje și distribuții de clase.

Rezultate și Constatări

Experimentele noastre pe Dronescapes demonstrează mai multe avantaje cheie ale metodei noastre. În Tabelul 5.1 arătăm că noile noastre **hiper-muchii mai complexe (AH, CH) depășesc constant performanța muchiilor mai simple de un singur salt și de două salturi**.

	Tip	Antrenare Neetichetat (iter 2)			Antrenare Neetichetat (iter 3)		
		(1)	(2)	(3)	(1)	(2)	(3)
Muchii	E: rgb	42.85	5.04	10.37	32.79	21.66	12.40
	E: hsv	41.70	4.69	10.54	33.51	19.90	12.48
	E: softedges	32.47	6.26	11.56	27.28	18.61	13.53
	E: softseg	30.71	5.97	11.14	24.68	22.70	12.76
	TH-E: sseg	-	6.25	11.39	-	19.00	12.93
	TH-E: depth	29.24	-	12.22	24.11	-	13.79
	TH-E: norm	30.56	6.17	-	26.35	21.15	-
Hiper-Muchii	AH	41.80	5.33	10.37	33.63	23.96	12.24
	CH	44.63	4.93	10.32	36.92	20.36	12.23

Table 5.1: Evaluarea muchiilor și Hiper-Muchiilor pentru task-uri multiple: 1 - segmentare semantică (sseg); 2 - estimarea adâncimii (depth); 3 - normalele suprafeței (norm).

În Tabelul 5.2, arătăm că ansamblurile noastre parametrice învățate îmbunătățesc semnificativ performanța față de metodele anterioare care folosesc medierea non-parametrică pe setul de date NEO.

Metodă	IoU(↑)			
	Barsana	Comana	Norway	Medie
NGC [18] (Medie)	41.53	40.75	27.38	36.55
NGC (Medie) + HE	42.61	42.17	27.96	37.58
CShift [14] (Medie)	43.91	42.13	29.68	38.57
CShift (Medie) + HE	44.71	43.88	30.09	39.56
LR (al nostru)	46.51	**45.59**	**30.17**	**40.76**
NN (v1) (al nostru)	45.53	42.92	28.37	38.94
NN (v2) (al nostru)	45.48	43.25	26.36	38.36
NN (v3) (al nostru)	**48.21**	44.85	28.94	40.67

Table 5.2: Ansambluri învățate comparate cu metodele existente.

În Tabelul 5.3, arătăm că procesul iterativ semi-supervizat duce la câștiguri substanțiale atât în acuratețe, cât și în coerența temporală pentru toate task-urile pe setul de date Dronescapes.

Tip	Semantic		Adâncime		Normale	
	IoU (↑)	Cons. (↑)	L1 (↓)	Cons. (↑)	L1 (↓)	Cons. (↑)
rgb-sup.	25.04	88.85	-	-	-	-
rgb-iter1	32.79	94.04	21.66	5.89	12.40	98.32
rgb-iter2	37.26	95.72	17.34	7.06	11.93	98.87
rgb-iter3	**40.31**	**98.13**	**16.64**	**30.26**	**11.71**	**99.30**

Table 5.3: Învățarea iterativă îmbunătățește constant atât acuratețea (IoU, L1), cât și coerența temporală (Cons.) pentru muchia principală *rgb* → *task* pe scenele de test.

În final, arătăm că modelul nostru poate prelua ieșirea unui expert puternic pre-antrenat (Mask2Former) și poate rafina în continuare predicțiile acestuia, îmbunătățind atât acuratețea, cât și coerența pe scene noi, nevăzute (Figura 5.6).

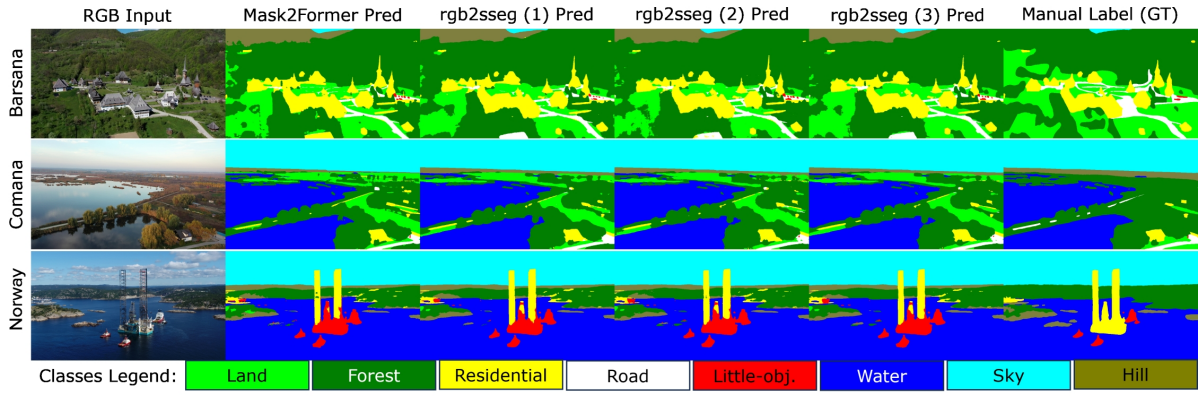


Figure 5.6: Rezultate calitative pe imaginile de test. Metoda noastră îmbunătățește labelurile de la o bază de referință puternică (Mask2Former), în special pe scenele din afara distribuției, cum ar fi Norvegia.

5.3 Analiză experimentală pe setul de date NEO

Set de date și Configurare Experimentală

Pentru a demonstra generalitatea abordării noastre, o aplicăm și la problema Observării Pământului folosind setul de date NASA NEO. Acest set de date conține 22 de ani de observații satelitare lunare pentru diverse straturi precum Temperatura Suprafeței Terestre, Adâncimea Optică a Aerosolilor și Indicele Suprafeței Foliare. Datele sunt provocatoare din cauza rarității și prezenței măsurătorilor lipsă pe perioade lungi, făcându-l un caz ideal pentru învățarea semi-supervizată. Folosim 12 straturi ca intrări pentru a prezice 7 straturi de ieșire diferite.

Observăm rezultate similare pe acest set de date în comparație cu cel Dronescapes, cum ar fi rezultate îmbunătățite folosind ansamblurile învățate, precum și rezultate îmbunătățite folosind învățarea iterativă semi-supervizată, așa cum este prezentat în Figura 5.7. Învățarea iterativă duce, de asemenea, la o creștere semnificativă a coerenței temporale, ceea ce înseamnă că predicțiile pentru o anumită locație sunt mai stabile și mai fiabile în timp.

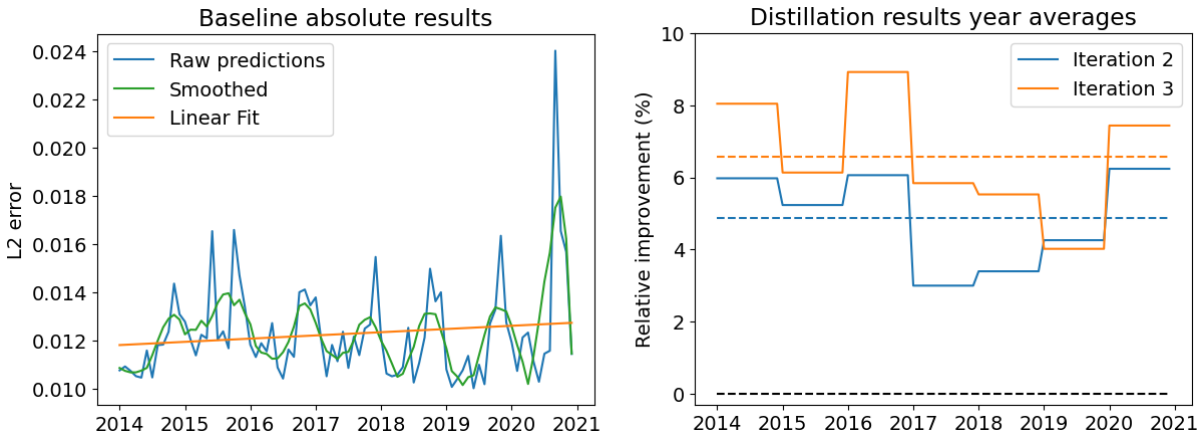


Figure 5.7: Erori de predicție pe parcursul a șapte ani. **Stânga**: Eroarea de bază arată o creștere graduală în timp, indicând o schimbare a distribuției datelor. **Dreapta**: Iterațiile noastre semi-supervizate reduc semnificativ această eroare, demonstrând capacitatea modelului de a se adapta.

5.4 Concluzii

În acest capitol, am introdus un model inovator de Hiper-Graf multi-task și semi-supervizat. Punctele sale forte principale sunt utilizarea de hiper-muchii complexe pentru a captura relații de ordin superior între task-uri și introducerea de ansambluri învățabile care acționează ca profesori pentru a ghida învățarea din date nelabeluite.

Am validat abordarea noastră pe două probleme reale foarte diferite și provocatoare: înțelegerea scenelor aeriene cu noul nostru set de date Dronescapes și observarea pe termen lung a Pământului cu setul de date NASA NEO. În ambele cazuri, metoda noastră a arătat îmbunătățiri semnificative în acuratețe, generalizare la date noi și coerență temporală, chiar și pornind de la date labeluite foarte limitate. Acest lucru demonstrează puterea și flexibilitatea cadrului de hiper-graf pentru învățarea semi-supervizată.

Chapter 6

Hiper-Grafuri Probabilistice folosind Autoencodere Mascate, ansambluri și distilare eficientă

Acest capitol rezumă progresele recente ale muncii mele privind hiper-grafurile neurale, ansamblurile și distilarea. Acesta se bazează pe capitolele anterioare prin introducerea unei noi metode probabilistice pentru definirea hiper-grafurilor folosind mascarea aleatorie într-un cadru multi-modal și multi-task. Setul de date Dronescapes, introdus anterior, este extins cu noi scene UAV și modalități intermediare derivate de la experți pre-antrenați.

Lumea reală este inerent multi-modală. Pentru o scenă exterioară din setul de date Dronescapes, putem deduce perspective distincte precum segmentarea semantică, adâncimea sau mișcarea. Sistemul nostru combină aceste perspective pentru a produce o înțelegere coerentă și unificată. Lucrările timpurii din această teză au modelat aceste relații folosind grafuri (Capitolul 4) și hiper-grafuri (Capitolul 5) cu o structură fixă. Abordăm această limitare cu o rețea neurală care învață interdependențele dintre perspective din date. Această abordare modelează distribuția tuturor configurațiilor posibile de hiper-graf, fiecare trecere de inferență eșantionând o configurație specifică.

Adaptăm sarcina proxy de autoencodare mascată (MAE) prin definirea unor seturi fixe de perspective de intrare și de ieșire și prin utilizarea unor funcții de pierdere specifice task-urii (de ex., entropie încrucișată pentru clasificare, L2 pentru regresie). Acest lucru unifică pașii de pre-antrenare și de fine-tuning într-o singură buclă. Mai mult, permite construirea de ansambluri la momentul inferenței prin căi aleatorii. Arătăm, de asemenea, că această cunoaștere poate fi distilată în modele foarte mici (sub 1M de parametri) cu o pierdere minimă de performanță. În această lucrare, îmbunătățim performanța prin mascarea unor perspective întregi, în loc de patch-uri, creând astfel hiper-noduri probabilistice la nivel de modalitate, ca în capitolul anterior. Numim metoda noastră Hiper-Grafuri Probabilistice, sau PHG-MAE.

Pentru a aborda dificultatea variabilă a prezicerii diferitelor task-uri, introducem modalități intermediare derivate. Acestea acționează ca o punte între intrările de nivel scăzut (cum

ar fi RGB) și ieșirile de nivel înalt (cum ar fi segmentarea), simulând o formă de învățare curriculară. Acest lucru, combinat cu abordarea noastră probabilistică, permite modelului să învețe dependențele optime intrare-ieșire direct din date.

O parte cheie a acestei cercetări a fost capacitatea de a itera rapid și de a adăuga noi perspective de la experți pre-antrenați. Pentru a facilita această cercetare, am dezvoltat o conductă de date (data pipeline) open-source pentru generarea automată de seturi de date din videoclipuri arbitrare, generând reprezentări multiple folosind experți pre-antrenați[1]. Deși lucrarea noastră se concentrează pe scene UAV exterioare, metodologia este aplicabilă și altor domenii, cum ar fi conducerea autonomă. Acest instrument a fost folosit pentru a extinde setul de date Dronescapes cu noi videoclipuri UAV augmentate cu cunoștințe de la experți, ceea ce a dus la o performanță mai bună în task-urile finale.

Pe scurt, principalele noastre contribuții sunt:

1. Hiper-Grafuri Probabilistice folosind Autoencodere Mascate (PHG-MAE): o extensie a algoritmului MAE standard care permite Ansambluri prin Mascare Aleatorie la momentul inferenței și unifică metodele anterioare de Hiper-Graf sub un singur model neuronal.
2. Fuzionarea pre-antrenării și a ajustării fine specifice task-urii într-o singură buclă de antrenament prin definirea intrărilor și ieșirilor diferit de bucla MAE standard. Mai mult, mascăm doar o întreagă perspectivă, nu la nivel de patch, așa cum se făcea anterior.
3. Includerea de modalități intermediare derivate de la experți pre-antrenați pe seturi de date diverse pentru a valorifica cunoștințele acestora și a netezi dificultatea de învățare de la intrări de nivel scăzut la task-uri complexe de nivel înalt.
4. Antrenare și distilare eficientă, demonstrând performanțe competitive cu rețele CNN mici (4.4M) și foarte mici (400k) pe setul de testare Dronescapes și pe videoclipuri nevăzute, permițând cercetarea pe hardware de consum.
5. O conductă de date open-source care permite extragerea eficientă de noi perspective de la experți pre-antrenați pe videoclipuri, simplificând procesul de antrenare a modelelor de viziune multi-task pe seturi de date video mari.

Seturi de date Extindem setul de date Dronescapes, introdus în Capitolul 5, cu modalități suplimentare și videoclipuri noi. Acest lucru crește numărul total de cadre adnotate de la 23K la 80K. Folosim conducta noastră de date pentru a augmenta setul de date cu noi perspective de la experți pre-antrenați, cum ar fi segmentarea semantică, estimarea adâncimii, fluxul optic și perspective derivate precum normalele camerei și hărțile de segmentare binară. Mai mult, setul nostru de date pentru distilare include un total de 148K de cadre.

Învățare Multi-modală Multi-task (MTL) Utilizăm o configurație de învățare multi-task în care perspectivele de intrare sunt denumite *modalități*, iar cele de ieșire *task-uri*. Experimentele noastre folosesc RGB ca modalitate de intrare primară și urmăresc să prezică trei task-uri de ieșire: segmentarea semantică, estimarea adâncimii și estimarea normalelor camerei.

[1] https://gitlab.com/video-representations-extractor/video-representations-extractor

Învățare prin ansamblu și ansambluri la momentul inferenței Învățarea prin ansamblu îmbunătățește performanța și consistența predicțiilor. În capitolele anterioare, am creat ansambluri folosind căi diferite într-un graf neuronal. Aici, generăm ansambluri valorificând aleatorismul din modelul nostru. Efectuăm mai multe treceri de inferență pentru aceeași intrare, de fiecare dată cu o mască aleatorie diferită aplicată modalităților de intrare. Spre deosebire de metodele anterioare care maschează patch-uri, noi mascăm perspective întregi, ceea ce corespunde eșantionării diferitelor hiper-muchii. Predicțiile rezultate sunt apoi agregate prin mediere simplă.

Învățare semi-supervizată În acest capitol, efectuăm și o singură iterație de învățare semi-supervizată prin distilare, deși accentul principal este pe modelul de hiper-graf probabilistic însuși.

6.1 PHG-MAE: Modelul de Hiper-Grafuri Probabilistice folosind Autoencodere Mascate

Principala limitare a hiper-grafurilor definite în capitolele anterioare a fost structura lor fixă, ceea ce le făcea dificil de modificat fără o reantrenare extinsă. Motivați de acest lucru, am formulat conceptul de hiper-graf în cadrul unei singure rețele neurale. Ca o contribuție teoretică, arătăm că un graf neuronal multi-strat este echivalent cu o singură rețea neurală mai mare. Prin mascarea corespunzătoare a intrărilor și a ponderilor interne ale acestei singure rețele, putem replica comportamentul mai multor rețele mai mici, independente, care formează muchiile unui graf. Demonstrăm că acest lucru este valabil pentru rețelele neurale convoluționale, așa cum este ilustrat în figurile de mai jos, iar principiul se generalizează la rețele adânci și alte tipuri de straturi. Această echivalență ne permite să reprezentăm un întreg hiper-graf într-un singur model.

Modificăm algoritmul MAE standard pentru a efectua simultan atât pre-antrenarea, cât și predicția specifică task-urii. Realizăm acest lucru cu două schimbări:

- Funcție de pierdere specifică task-urii. Utilizăm funcții de pierdere adecvate pentru fiecare task de ieșire, cum ar fi entropia încrucișată pentru segmentarea semantică și pierderea L2 pentru task-urile de regresie, cum ar fi estimarea adâncimii.
- Definirea intrărilor și ieșirilor. Definim două seturi disjuncte de perspective. Perspectivele de intrare (de ex., RGB) sunt ușor de achiziționat și sunt **întotdeauna văzute** de model. Perspectivele de ieșire (de ex., segmentarea semantică) sunt greu de achiziționat și sunt **întotdeauna mascate**, forțând modelul să le prezică. Acest lucru combină paradigma auto-encoderului cu învățarea supervizată standard.

Ansambluri prin Mascare Aleatorie

În loc să creăm manual aceste măști, lăsăm modelul să învețe interdependențele dintre perspective folosind principiile Autoencoderelor Mascate (MAE). Tratăm fiecare perspectivă de imagine completă ca pe un singur token care urmează să fie mascat.

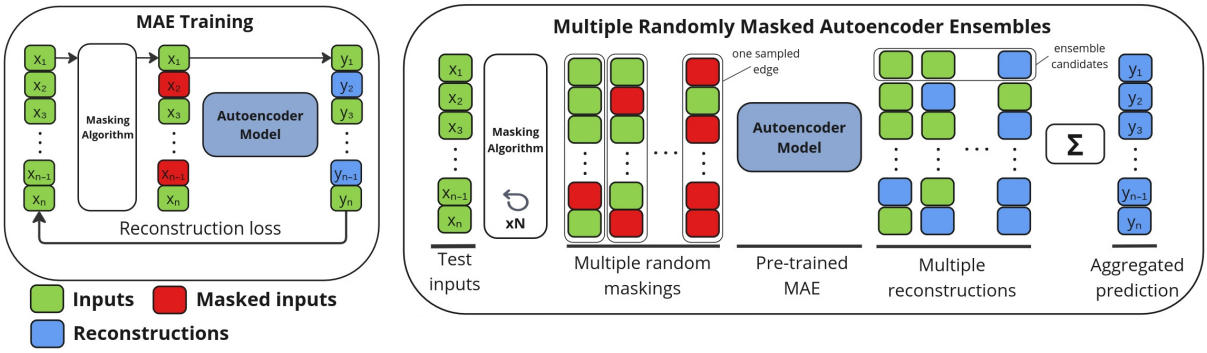


Figure 6.1: Stânga: Antrenarea unui model MAE standard. Dreapta: Ansambluri prin Mascare Aleatorie la momentul testării folosind modelul antrenat.

Așa cum se arată în Figura 6.1, în timpul antrenamentului, mascăm aleatoriu unele perspective și însărcinăm rețeaua cu reconstruirea lor din cele vizibile. La momentul testării, putem interoga modelul antrenat de mai multe ori pentru aceeași intrare, de fiecare dată cu o mască aleatorie diferită. Predicțiile colectate sunt apoi agregate pentru a forma un ansamblu.

Propoziție O trecere înainte (forward pass) printr-un Autoencoder Mascat este echivalentă cu o trecere înainte a unei singure muchii într-un Hiper-Graf. Prin mascarea aleatorie eșantionăm din distribuția tuturor hiper-muchiilor.

Modalități intermediare pentru diversitatea ansamblului

Dacă intrările sunt întotdeauna văzute și ieșirile sunt întotdeauna mascate, nu există aleatorism pentru a genera ansambluri la momentul testării. Pentru a rezolva acest lucru, introducem un set de modalități intermediare. Acestea sunt derivate de la experți pre-antrenați folosind conducta noastră de date, așa cum se arată în Figura 6.2. Aceste modalități, care includ predicții de la experți pentru adâncime și segmentare, precum și perspective derivate mai simple, cum ar fi hărți binare pentru „vegetație" sau „cer", sunt **uneori mascate și uneori văzute** în timpul antrenamentului. La inferență, mascarea aleatorie a acestor modalități intermediare ne permite să generăm un set divers de predicții pentru ansamblare. Aceste perspective servesc, de asemenea, ca o punte între intrările de nivel scăzut și task-urile complexe de nivel înalt, ajutând procesul de învățare.

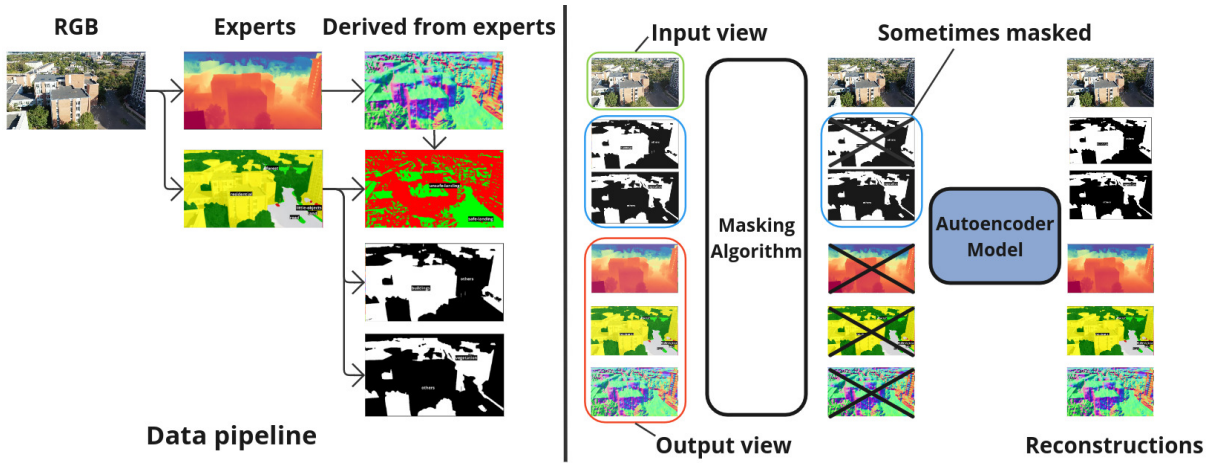


Figure 6.2: Conducta de Date (Data-Pipeline) și modelul PHG-MAE pe date reale. Stânga: Derivarea modalităților de la experți pre-antrenați folosind doar RGB. Dreapta: Integrarea modalităților în model.

6.2 Analiză experimentală

Descrierea setului de date

Toate experimentele sunt evaluate pe setul de date de referință *Dronescapes*. Extindem setul de date original cu 8 noi scene video UAV și numeroase modalități extrase cu conducta noastră de date. Acest lucru duce la variantele *Dronescapes-Ext* și *Dronescapes-*-M*, care cresc semnificativ datele de antrenament de la 23K la 80K de cadre. Tabelul 6.1 rezumă variațiile setului de date. Modelele noastre sunt antrenate eficient pe hardware de consum, cel mai mare model al nostru fiind antrenat în puțin sub o săptămână, în timp ce cel mai mare model de distilare în aproximativ două zile.

Nume	Puncte de date (labeluri GT)	Intrări/Ieșiri Perspective	Scene UAV	Descriere
Dronescapes-Train	12K (233)	5/3	7	Setul de antrenament original
Dronescapes-Semisup	11K (207)	5/3	7	Setul semi-supervizat original
Dronescapes-Test	5.6K* (116)	5/3	3	Setul de testare original
Dronescapes-Pseudo	23K (233)	5/3	7	Pseudo-labeluri din Dronescapes-Semisup
Dronescapes-Train-M	12K (233)	14/3	7	Toți experții noi și modalitățile intermediare
Dronescapes-Ext	80K (440)	1/3	15	Primele două seturi combinate plus conducta de date pe videoclipuri noi (fără modalități noi)
Dronescapes-Ext-M	80K (440)	14/3	15	Toți experții noi și modalitățile intermediare pe setul de date extins
Dronescapes-Ext2-Pseudo	148K (440)	1/1	23	Pseudo-labeluri pe Dronescapes-Ext plus 8 videoclipuri UAV noi

Table 6.1: Variații și statistici ale setului de date Dronescapes. Numerele din paranteze reprezintă datele semantice adnotate de oameni.

Rezultate privind Învățarea Multi-Task

Comparăm modelul nostru PHG-MAE cu modelele de referință din capitolele anterioare pe cele trei task-uri principale ale setului de date Dronescapes. Așa cum se arată în Tabelul 6.2, modelele noastre, în special atunci când sunt antrenate pe setul de date extins cu modalități suplimentare, ating performanțe de ultimă generație, depășind semnificativ metodele anterioare precum NGC și SafeUAV-MTL. Rezultatele calitative din Figura 6.3 confirmă că modelul nostru se generalizează bine la scene de test nevăzute.

Model	Antrenament Set de date	Parametri	Semantic ↑ Segmentare	Adâncime ↓ Estimare	Normale Cameră Estimare ↓
PHG-MAE-MTL	Dronescapes-Ext	4.4M	52.04	18.84	12.67
PHG-MAE	Dronescapes-Ext-M	4.4M	49.09 ± 3.8	19.57 ± 2.2	13.68 ± 1.7
PHG-MAE	Dronescapes-Train-M	4.4M	42.84 ± 4.1	18.23 ± 1.5	12.54 ± 1.5
PHG-MAE-MTL	Dronescapes-Train	1.1M	39.23	19.31	13.18
PHG-MAE-MTL	Dronescapes-Train	4.4M	39.1	20.55	13.48
NGC-HE(mean) [20]	Dronescapes-Pseudo	32M	37.58	21.81	12.40
NGC(mean) [18]	Dronescapes-Pseudo	32M	36.55	20.08	12.97
SafeUAV-MTL [19]	Dronescapes-Train	1.1M	32.79	21.66	12.40

Table 6.2: Comparație privind învățarea multi-task.

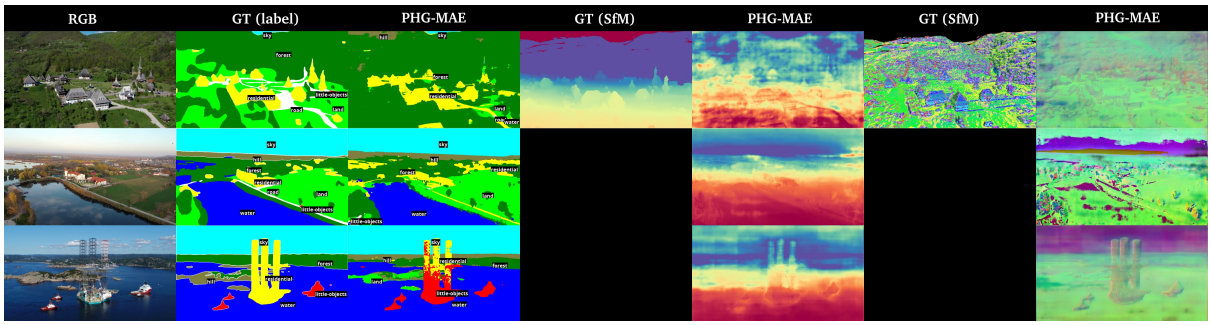


Figure 6.3: Rezultate calitative pentru Învățarea Multi-Task (MTL) ale celui mai bun model al nostru din Tabelul 6.2. Etichetele de referință pentru Segmentarea Semantică sunt adnotate uman, în timp ce pentru Adâncime și Normalele Camerei, acestea se bazează pe o reconstrucție prin Structură din Mișcare. Doar o singură scenă este disponibilă în setul de testare original Dronescapes pentru aceste două task-uri.

Rezultate privind Segmentarea Semantică

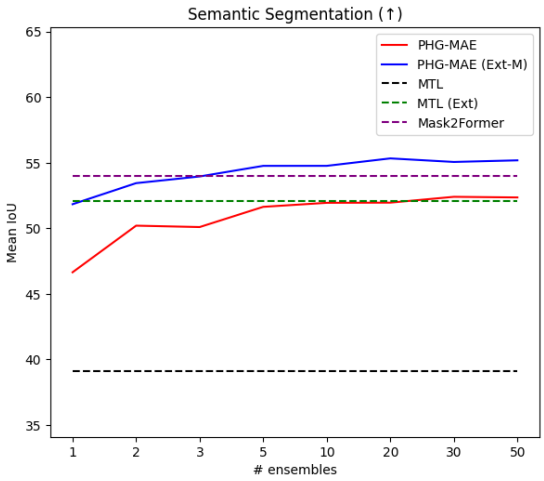
Concentrându-ne pe segmentarea semantică, care are date de referință fiabile adnotate de oameni, cel mai bun model al nostru antrenat pe setul de date extins atinge un IoU Mediu de 52.04. Așa cum se arată în Tabelul 6.3, acest rezultat depășește lucrările anterioare și este competitiv cu Mask2Former, un model mult mai mare bazat pe transformatoare.

Model	Set de Date Antrenament	Parametri	IoU Mediu ↑
Mask2Former[15]	Mapillary[21]	216M	53.97
PHG-MAE-MTL	Dronescapes-Ext	4.4M	52.04
PHG-MAE	Dronescapes-Ext-M	4.4M	51.83±3.3
PHG-MAE	Dronescapes-Train-M	4.4M	46.64±5.1
NGC-HE-LR[20]	Dronescapes-Pseudo	32M	40.76
SafeUAV[19]	Dronescapes-Pseudo	1.1M	40.31
PHG-MAE-MTL	Dronescapes-Train	1.1M	39.23
PHG-MAE-MTL	Dronescapes-Train	4.4M	39.1
NGC-mean[18]	Dronescapes-Pseudo	32M	36.55
SafeUAV[19]	Dronescapes-Train	1.1M	32.79

Table 6.3: Evaluarea segmentării semantice pe setul de testare Dronescapes.

Rezultate privind Învățarea prin Ansambluri

Ansamblurile noastre prin Mascare Aleatorie oferă o creștere substanțială a performanței. Așa cum se arată în Figura 6.4, agregarea predicțiilor de la mai multe măști aleatorii la momentul testării îmbunătățește scorul de segmentare semantică al celui mai bun model al nostru de la 51.83 la 55.32 IoU Mediu. Acest rezultat de ansamblu depășește performanța expertului Mask2Former de 216M parametri. Ansamblurile produc, de asemenea, predicții mai stabile și calitativ mai bune pe scene nevăzute, așa cum se arată în Figura 6.5.



Model	IoU Mediu ↑ (fără ansambluri)	IoU Mediu ↑ (cu ansambluri)
PHG-MAE (Ext-M)	51.83±3.3	$\mathbf{55.32 \pm 0.54}$
Mask2Former	**53.97**	n/a
MTL (Ext)	52.04	n/a
PHG-MAE	46.64±5.1	52.26±0.54
MTL	39.1	n/a

Figure 6.4: Rezultate pentru Învățarea Multi-Task (MTL) cu ansambluri față de modelul de referință. Stânga: Grafic cu diverse rezultate intermediare pentru un număr mic de candidați ai ansamblului. Dreapta: Cea mai bună predicție unică (fără ansamblu) și cea mai bună predicție de ansamblu (50 de candidați).

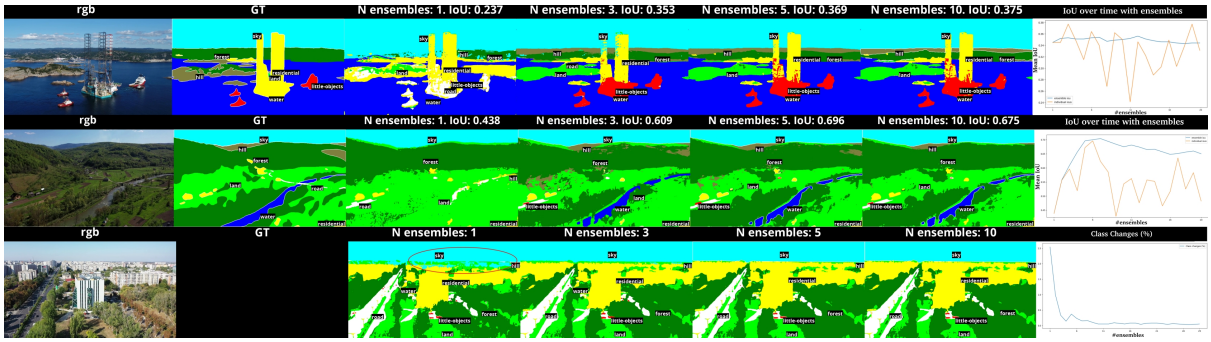


Figure 6.5: Ansambluri prin Mascare Aleatorie pe scene de testare nevăzute anterior.

Rezultate privind Distilarea

Pentru a crea un model practic pentru implementare, distilăm cunoștințele de la modelul nostru complex, multi-modal și de ansamblu într-o rețea CNN simplă și ușoară, care necesită doar intrare RGB. Așa cum este detaliat în Tabelul 6.4, acest proces este foarte eficient. Un model distilat cu doar 430k de parametri atinge un IoU Mediu de 54.94, păstrând aproape toată performanța profesorului său mare, fiind în același timp cu ordine de mărime mai rapid și mai ușor de implementat, așa cum se vede în Tabelul 6.4.

Model	Parametri	Nr. Intrări Modalități	Conductă de Date	Ansambluri Aleatorii	Date Suplimentare	IoU Mediu ↑	Timp de rulare (s) ↓
PHG-MAE	4.4M	12	✓	✓	✗	55.32±0.54	78.9
PHG-MAE-Distil	4.4M	1	✗	✗	✓	55.05	0.064
PHG-MAE-Distil	430k	1	✗	✗	✓	54.94	0.054
PHG-MAE-Distil	4.4M	1	✗	✗	✗	54.37	0.064
PHG-MAE-Distil	1.1M	1	✗	✗	✓	54.3	0.058
Mask2Former [6]	217M	1	✗	✗	✗	53.97	0.79
PHG-MAE-Distil	150k	1	✗	✗	✓	53.32	0.052
PHG-MAE-Distil	430k	1	✗	✗	✗	52.44	0.054
PHG-MAE-MTL	4.4M	1	✗	✗	✗	52.04	0.064
PHG-MAE	4.4M	12	✓	✗	✗	51.83±3.3	74.4

Table 6.4: Rezultate de Distilare pentru Învățare cu o Singură Task peste ansamblurile PHG-MAE.

Coerență temporală Evaluăm, de asemenea, coerența temporală folosind o metrică bazată pe alinierea fluxului optic între cadre consecutive. Modelele noastre distilate nu sunt doar precise, ci produc și predicții foarte consistente în timp, depășind atât profesorul de ansamblu, cât și modelul de referință mare Mask2Former. Acest lucru le face potrivite pentru aplicații robotice din lumea reală, unde stabilitatea temporală este crucială.

6.3 Concluzii

Introducem un nou algoritm de ansamblu la momentul testării care funcționează pe orice model de tip Autoencoder Mascat, care este un mecanism de pre-antrenament foarte popular astăzi pentru modelele mari. Am testat algoritmul nostru pe modele mici bazate pe CNN cu 1.1M și 4.4M de parametri pe setul de date Dronescapes, cu modele antrenate pe hardware de consum. Arătăm că ansamblurile la momentul testării depășesc paradigma clasică de predicție a Învățării Multi-Task, producând hărți de segmentare semantică de o calitate superioară și mai consistente, chiar și cu metoda simplă de ansamblare prin mediere. Acest lucru deschide,

de asemenea, calea pentru cercetări viitoare, cum ar fi metode de agregare mai bune, precum utilizarea unei rețele neurale secundare pentru agregare directă sau căutarea și filtrarea candidaților.

Chapter 7

Concluzii și direcții viitoare de cercetare

Acest capitol final prezintă o imagine de ansamblu a tezei, reflectând asupra ideilor sale de bază și conturând direcții interesante pentru cercetări viitoare.

Lucrarea se situează la intersecția mai multor domenii, inclusiv *învățare adâncă* cu *rețele neurale*, *învățare multi-task*, *consensul predicțiilor prin învățare prin ansamblu*, *distilarea modelelor*, *grafuri* și *învățare semi-supervizată*. Am aplicat aceste concepte în robotică, în special pentru *înțelegerea scenelor aeriene* cu UAV-uri și date de observare a Pământului.

Capitolele tezei sunt concepute pentru a se construi unul pe celălalt. În Capitolul 3, am explorat învățarea prin ansamblu și distilarea pentru a îmbunătăți estimarea adâncimii. În Capitolele 4 și 5, am folosit grafuri și rețele neurale pentru a modela lumea multi-modală, introducând un nou algoritm de învățare semi-supervizată. În final, în Capitolul 6, am distilat modelul complex de graf într-o singură rețea neurală folosind mascarea aleatorie, conectând lucrarea noastră cu metodele de pre-antrenament de ultimă generație.

O concluzie cheie a acestei cercetări este importanța partajării muncii noastre. Făcând ideile, codul, experimentele și rezultatele disponibile în mod deschis, îi ajutăm pe alții să construiască pe descoperirile noastre și să accelereze progresul științific. Acest proces de partajare deschisă și de a permite altora să *distileze* lucrările anterioare este crucial pentru comunitate.

Direcții Viitoare de Cercetare

Cercetarea prezentată aici deschide mai multe direcții promițătoare pentru explorări viitoare.

Înțelegerea scenei Deși această teză s-a concentrat pe înțelegerea scenelor aeriene, domeniul este mult mai larg. Lucrările viitoare ar putea integra tehnici moderne de reconstrucție 3D precum NeRF [9] și 3D Gaussian Splatting [4]. O altă cale interesantă este de a combina metodele bazate pe date cu abordări care modelează legile fizice, cum ar fi Rețelele Neurale Informate de Fizică [7] sau Modelele Lumii [32], pentru a obține o înțelegere mai holistică a unei scene.

Rețele neurale Am folosit în principal arhitectura SafeUAV [19] datorită eficienței sale. Un

pas următor evident este explorarea integrării unor arhitecturi mai recente, cum ar fi Transformatoarele [30], care au devenit standard în multe domenii. O altă direcție este utilizarea tehnicilor precum AutoML [27] pentru a descoperi automat arhitecturi de rețea optimizate pentru un set specific de task-uri.

Învățare adâncă și raționament Modelele actuale de învățare adâncă sunt adesea antrenate pe seturi de date mari și redundante. Lucrările viitoare ar putea explora tehnici de selecție a seturilor de date [1] pentru a identifica cele mai informative puncte de date pentru antrenament. Mai mult, există un interes crescând în depășirea simplei recunoașteri a tiparelor către raționament. Aceasta implică utilizarea unor metode precum lanțul de gândire [31] sau sinteza de programe [2] pentru a descompune problemele complexe în părți mai mici, rezolvabile.

Învățare multi-modală și multi-task prin ansamblu și grafuri Acest subiect a fost central în teză. Procesul nostru iterativ de învățare a muchiilor grafului, aplicarea ansamblurilor și apoi distilarea ar putea fi integrat într-o singură buclă de antrenament end-to-end. Am putea, de asemenea, să extindem modelul nostru pentru a utiliza tipuri de hiper-muchii mai complexe (vezi Secțiunea 5.1) și să încorporăm aceste ansambluri învățate în abordarea cu o singură rețea dezvoltată în Capitolul 6.

Învățare nesupervizată și semi-supervizată Această teză s-a bazat în mare parte pe învățarea supervizată. Cu toate acestea, progrese recente au arătat puterea unei abordări în două etape: mai întâi, pre-antrenarea unui model pe seturi de date mari, nelabeluite, folosind algoritmi nesupervizați, și apoi fine-tuning pe task-uri specifice, labeluite. Adoptarea acestei paradigme ar fi o extindere valoroasă a muncii noastre.

Integrare și implementare în sisteme reale O direcție de interes personal este integrarea acestor modele de învățare adâncă în sisteme robotice reale. În acest context, învățarea automată servește ca o componentă de percepție în cadrul unui sistem mai mare care trebuie, de asemenea, să *acționeze* asupra mediului său. Acest amestec de cod tradițional și modele învățate este uneori denumit Software 2.0 [16] și reprezintă un pas critic în aducerea cercetării în aplicații practice.

Acestea sunt doar câteva idei pentru ceea ce ar putea urma. Cheia este să urmărim o muncă ce inspiră curiozitate și plăcere. Vă mulțumesc pentru lectură.

Bibliography

[1] Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024.

[2] Shraddha Barke, Emmanuel Anaya Gonzalez, Saketh Ram Kasibatla, Taylor Berg-Kirkpatrick, and Nadia Polikarpova. Hysynth: Context-free llm approximation for guiding program synthesis. *Advances in Neural Information Processing Systems*, 37:15612–15645, 2024.

[3] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[4] Guikun Chen and Wenguan Wang. A survey on 3d gaussian splatting. *arXiv preprint arXiv:2401.03890*, 2024.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[6] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.

[7] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3): 88, 2022.

[8] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[9] Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*, 2022.

[10] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3205–3214, 2019.

[11] Yuan Gao, Haoping Bai, Zequn Jie, Jiayi Ma, Kui Jia, and Wei Liu. Mtl-nas: Task-agnostic neural architecture search towards general-purpose multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11543–11552, 2020.

[12] Google. Google earth, 2018. URL https://www.google.com/earth/. Available at https://www.google.com/earth/, version 7.3.0.

[13] Carsten Griwodz, Simone Gasparini, Lilian Calvet, Pierre Gurdjos, Fabien Castan, Benoit Maujean, Gregoire De Lillo, and Yann Lanthony. Alicevision Meshroom: An open-source 3D reconstruction pipeline. In *Proceedings of the 12th ACM Multimedia Systems Conference - MMSys '21*. ACM Press, 2021. doi: 10.1145/3458305.3478443.

[14] Emanuela Haller, Elena Burceanu, and Marius Leordeanu. Self-supervised learning in multi-task graphs through iterative consensus shift. *arXiv preprint arXiv:2103.14417*, 2021.

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[16] Andrej Karpathy. Software 2.0. https://web.archive.org/web/20250323195948/https://karpathy.medium.com/software-2-0-a64152b37c35, 2025. [Online; accessed 04-April-2025].

[17] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[18] Marius Leordeanu, Mihai Cristian Pîrvu, Dragos Costea, Alina E Marcu, Emil Slusanschi, and Rahul Sukthankar. Semi-supervised learning for multi-task scene understanding by neural graph consensus. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1882–1892, 2021.

[19] Alina Marcu, Dragos Costea, Vlad Licaret, Mihai Pîrvu, Emil Slusanschi, and Marius Leordeanu. Safeuav: Learning to estimate depth and safe landing areas for uavs from synthetic data. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[20] Alina Marcu, Mihai Pirvu, Dragos Costea, Emanuela Haller, Emil Slusanschi, Ahmed Nabil Belbachir, Rahul Sukthankar, and Marius Leordeanu. Self-supervised hypergraphs for learning multiple world interpretations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 983–992, 2023.

[21] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017.

[22] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12674–12684, 2020.

[23] Mihai Pirvu, Victor Robu, Vlad Licaret, Dragos Costea, Alina Marcu, Emil Slusanschi, Rahul Sukthankar, and Marius Leordeanu. Depth distillation: unsupervised metric depth estimation for uavs by finding consensus between kinematics, optical flow and deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3215–3223, 2021.

[24] Mihai Pirvu, Alina Marcu, Maria Alexandra Dobrescu, Ahmed Nabil Belbachir, and Marius Leordeanu. Multi-task hypergraphs for semi-supervised learning using earth observations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3404–3414, 2023.

[25] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[27] Imrus Salehin, Md Shamiul Islam, Pritom Saha, SM Noman, Azra Tuni, Md Mehedi Hasan, and Md Abu Baten. Automl: A systematic review on automated machine learning with neural architecture search. *Journal of Information and Intelligence*, 2(1):52–81, 2024.

[28] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[29] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419. Springer, 2020.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[31] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[32] Zheng Zhu, Xiaofeng Wang, Wangbo Zhao, Chen Min, Nianchen Deng, Min Dou, Yuqi Wang, Botian Shi, Kai Wang, Chi Zhang, et al. Is sora a world simulator? a comprehensive survey on general world models and beyond. *arXiv preprint arXiv:2405.03520*, 2024.