

1. Fase di studio preliminare

1.1. Kan applicate al field segmentation (Baralis Polito)

Il paper dal titolo “KAN You See It? KANs and Sentinel for Effective and Explainable Crop Field Segmentation” arXiv:2408.07040v1 mostra che:

Il lavoro propone l'integrazione dei nuovi Kolmogorov–Arnold Networks (KAN) all'interno dell'architettura U-Net (ottenendo U-KAN) per la segmentazione automatica dei campi coltivati a partire da immagini satellitari Sentinel-1 e Sentinel-2. L'obiettivo è duplice: migliorare l'accuratezza e l'efficienza del modello (minori GFLOPs) rispetto alla classica U-Net e fornire spiegazioni interpretabili (saliency maps) sul processo decisionale della rete .

Metodologia

- **Dataset:** South Africa Crop Type, immagini 256×256 sia multispettrali (12 canali Sentinel-2) sia radar (2 canali VV/VH di Sentinel-1).
- **Architetture:** U-Net standard vs. U-KAN (sostituisce il “cuore” dell’U-Net con blocchi Tok-KAN che apprendono funzioni di attivazione spline sui pesi).
- **Addestramento:** AdamW, loss di dice generalizzato, 60 epoch, data augmentation con ribaltamenti orizzontali/verticali.
- **Metriche:** IoU, F1, Precision, Recall e GFLOPs per valutare accuratezza e costo computazionale; Grad-CAM per spiegabilità post-hoc .

Risultati principali

- **Performance:**
 - Su Sentinel-2 U-KAN raggiunge un IoU del 74.8% contro il 72.9% di U-Net, con circa la metà dei GFLOPs (45.6 vs. 80.6) e un guadagno di precision di $\approx+3\%$.
 - Su Sentinel-1 le performance in IoU sono paragonabili (65.4% vs. 65.6%), ma U-KAN mantiene la maggiore efficienza computazionale.
- **Spiegabilità:**
 - Le saliency map di U-KAN si concentrano soprattutto sui bordi dei campi coltivati, mentre U-Net tende a focalizzarsi sulle aree interne. U-KAN mostra maggiore “plausibility” (IoU tra mappa di calore e ground truth: 73.5% vs. 68.2%) e, occludendo singoli canali Sentinel-2, evidenzia l'importanza cruciale delle bande B05, B8A e B11 per la segmentazione .

In sintesi, U-KAN migliora accuratezza e interpretabilità rispetto a U-Net, riducendo al contempo il carico computazionale e offrendo spiegazioni visive più focalizzate e plausibili.

1.2. UKAN

Ecco un breve riassunto del paper “U-KAN Makes Strong Backbone for Medical Image Segmentation and Generation”:

- Obiettivo e metodo

Gli autori propongono U-KAN, una variante di U-Net che integra strati di Kolmogorov-Arnold Networks (KAN) nella parte encoder-decoder. In particolare, dopo alcuni blocchi convoluzionali iniziali, le feature vengono “tokenizzate” in patch e processate da blocchi Tok-KAN—ciascuno composto da uno strato KAN seguito da convoluzioni depth-wise, normalizzazione e residuo—prima di ricostruire la mappa di segmentazione tramite skip-connection tipiche di U-Net .

- Risultati di segmentazione

Su tre dataset eterogenei (BUSI, GlAS, CVC), U-KAN supera tutti i confronti con U-Net classico, U-Net++, Att-Unet, U-NeXt, Rolling-UNet e U-Mamba, ottenendo in media un IoU di 78.7% e un F1 di 87.2%, con un costo computazionale inferiore (14.0 Gflops) e parameter size moderato (6.35 M) .

- Applicazione generativa

Sostituendo il backbone di un modello diffusion con U-KAN (Diffusion U-KAN), gli autori riportano una riduzione del Fréchet Inception Distance (da ~105 a ~102 sul dataset BUSI) e un miglioramento dell’Inception Score (fino a 2.76), dimostrando l’efficacia di U-KAN anche nella generazione di immagini mediche .

- Ablation e interpretabilità

Studi di ablation mostrano che l’uso di tre strati KAN è ottimale e che sostituirli con MLP tradizionali degrada le prestazioni. Inoltre, le mappe di attivazione dei KAN layer evidenziano una maggiore capacità di evidenziare regioni di interesse, confermando l’incremento di interpretabilità introdotto dai KAN .

In sintesi, U-KAN combina vantaggi teorici delle KAN (modellazione non lineare e spiegabilità) con l’efficienza e l’accuratezza richieste dai task visivi, risultando superiore sia nei benchmark di segmentazione che in quelli generativi.

1.3. FloodKAN

“FloodKAN: Integrating Kolmogorov–Arnold Networks for Efficient Flood Extent Extraction”:

Contesto e obiettivi

Gli autori partono dall’esigenza di migliorare l’estrazione delle aree inondate da immagini SAR, superando i limiti dei metodi tradizionali (soglie, SVM, RF) e le difficoltà dovute a rumore e terreni complessi. Propongono un modello denominato **UKAN**, che integra blocchi U-Net con layer basati sulla rete di Kolmogorov–Arnold (KAN), capaci di apprendere funzioni

di attivazione unidimensionali adattive per catturare relazioni non lineari e caratteristiche globali in modo più efficiente .

Metodologia

- **Architettura UKAN:** struttura a U (encoder-decoder) in cui ai classici blocchi convoluzionali si alternano moduli KAN;
- **Input multisorgente:** fusione di immagini Sentinel-1 dual-polarizzazione (VV, VH), modello digitale del terreno (DEM) e mappe storiche di distribuzione delle acque;
- **Addestramento:** su due dataset pubblici (C2S-MS Floods: 675 train / 75 val; MMFlood: 4269 train / 474 val), utilizzando Adam ($\text{lr}=3\text{e-}4$), batch size 16, 100 epoch .

Risultati principali

- **C2S-MS Floods:** UKAN ottiene un IoU del 87,95%, contro 84,46% di U-Net, 84,33% di Trans U-Net e 84,46% di Attention U-Net (miglioramento di ~3,5 punti). Precisione e recall raggiungono rispettivamente il 93,87% e 94,33% ($F1 = 93,55\%$) .
- **MMFlood:** IoU = 78,31% versus 75,24% (U-Net), 76,93% e 77,66% (altri), con $F1 = 87,75\%$, in miglioramento di circa 3 punti .
- **Analisi di complessità:** UKAN usa 37,54 M di parametri, sensibilmente meno di Trans U-Net e Attention U-Net, ma con performance superiori; evidenziata la migliore efficienza complessiva .

Conclusioni

UKAN combina efficacemente estrazione locale (convoluzioni) e modellazione globale (KAN), garantendo maggiore accuratezza e robustezza nell'estrazione delle aree alluvionate, con un buon compromesso tra prestazioni e complessità computazionale. Questi risultati sostengono l'impiego di UKAN in sistemi di monitoraggio e allerta rapida di eventi di piena.

1.4. Fields of the world

Ecco un riassunto conciso del paper:

2. Obiettivo

Introdurre **Fields of The World (FTW)**, un nuovo dataset di benchmark per la segmentazione automatica dei confini dei campi agricoli a livello globale, capace di supportare metodi di machine learning su paesaggi molto diversi (24 Paesi in 4 continenti) .

3. Caratteristiche del dataset

- 3.1. **Dimensione e copertura:** 70 462 “chip” da 1 536 m × 1 536 m (in totale 166 293 km²) con 1,63 M di poligoni di campo.
- 3.2. **Input:** immagini Sentinel-2 multispettrali (RGB + near-infrared), su due finestre temporali (stagione vegetativa e post-raccolto).
- 3.3. **Annotazioni:** maschere di segmentazione semantica (2-classi e 3-classi) e istanza (polygon) in formato interoperabile (fibia).

4. Esperimenti di baseline

- 4.1. Modello principale: **U-Net** con backbone EfficientNet-b3, addestrato su maschere a 3 classi (interno, bordo, sfondo), ignorando i pixel “sconosciuti” nei paesi con etichette parziali.
- 4.2. **Risultati di valutazione** (su tre paesi di esempio):
 - 4.2.1. Slovenia: Pixel IoU ≈ 0.59
 - 4.2.2. Francia: Pixel IoU ≈ 0.79
 - 4.2.3. Sud Africa: Pixel IoU ≈ 0.80 .

5. Zero-shot e transfer learning

- 5.1. Modelli pre-addestrati su FTW, senza alcuna messa a punto locale, mostrano prestazioni paragonabili o superiori a quelli pre-addestrati su dataset geograficamente più limitati (es. AI4Boundaries) e poi fine-tuned sul paese target.
- 5.2. Applicazione **zero-shot** in Etiopia (paese non incluso in FTW): buone predizioni qualitative dei confini dei campi, suggerendo elevata generalizzabilità .

6. Conclusioni

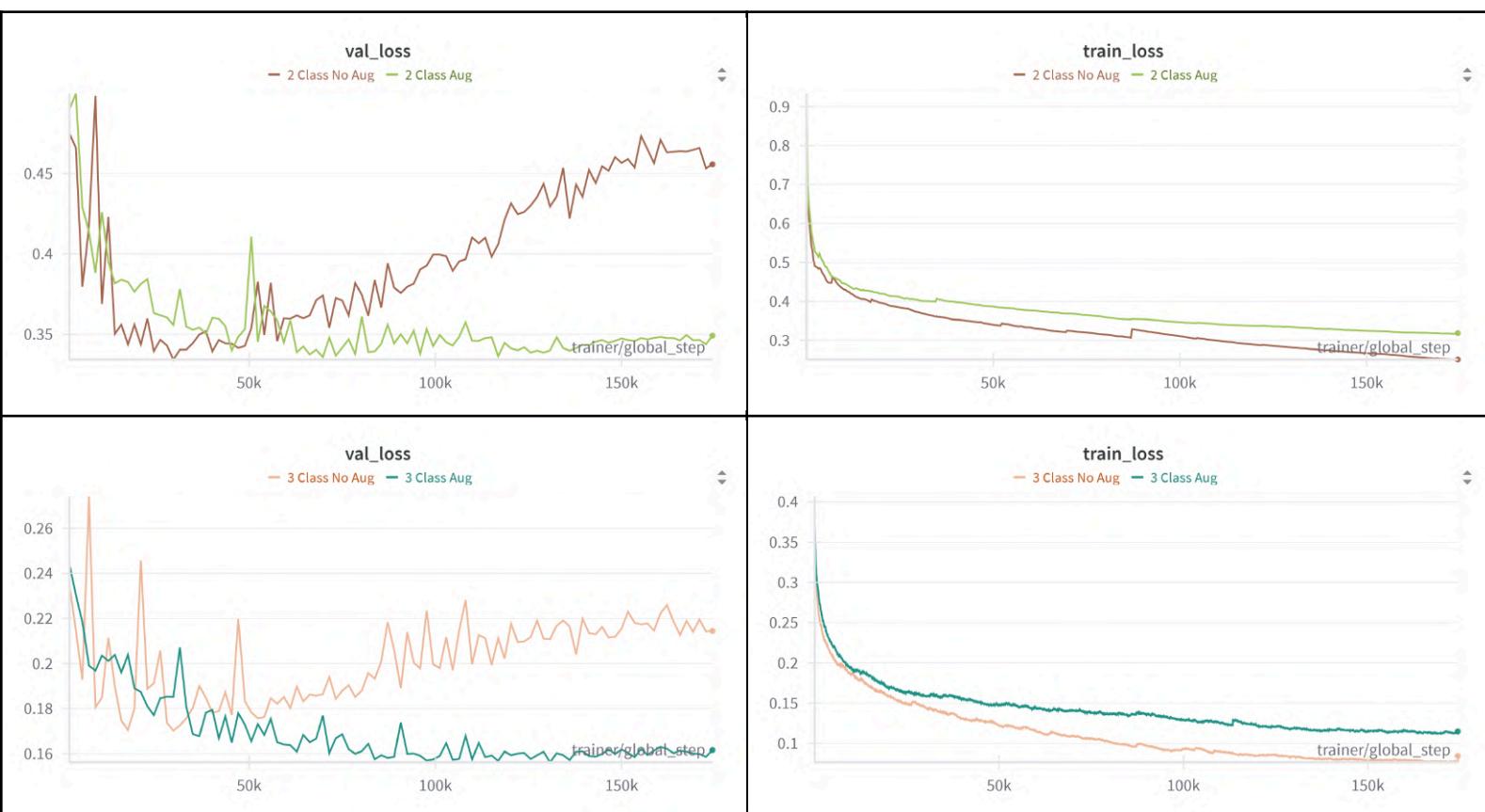
FTW offre un dataset senza precedenti per ricercatori e operatori nel settore agricolo, permettendo di sviluppare e valutare modelli di segmentazione di confini di campi agricoli su scala globale e con diversità morfologica e climatica inedite.

- 6.1.1. **Tipologie di campi**
- 6.1.2. **Modalità di acquisizione**
- 6.1.3. **Normalizzazione delle immagini**

7. Fase di replicazione

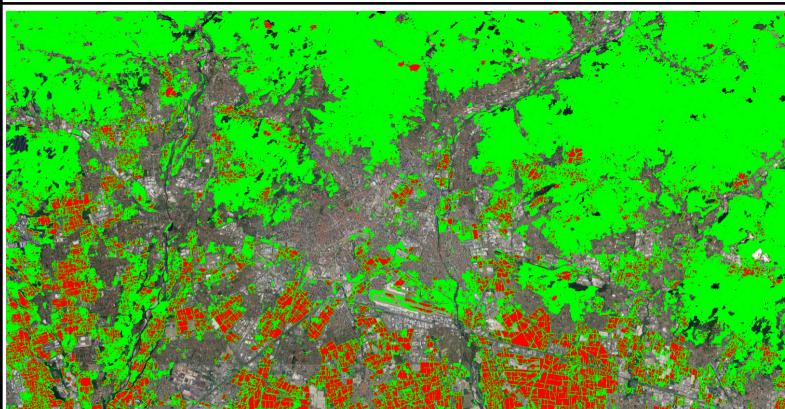
- 7.1. **Architettura ResNet + Unet di FOTW**
 - 7.1.1. **Risultati in accordo con paper**

MODEL	test_countries	pixel_level_iou	pixel_level_precision	pixel_level_recall	object_level_precision	object_level_recall	Trainable Parameters
FTW 2C (0mp6si3n)	all (NO PO)	0.8210	0.8691	0.9368	0.3874	0.1389	13.2 M
FTW 2C AUG (x2cleaoi)	all (NO PO)	0.8226	0.8643	0.9446	0.4009	0.1168	13.2 M
FTW 3C (puvs09ur)	all (NO PO)	0.7863	0.9198	0.8441	0.4088	0.3745	13.2 M
FTW 3C AUG (pjfrxtfj)	all (NO PO)	0.7662	0.9205	0.8205	0.3942	0.3354	13.2 M

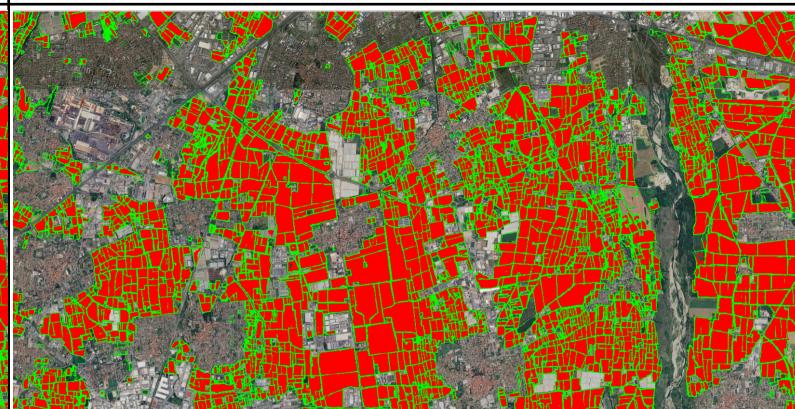
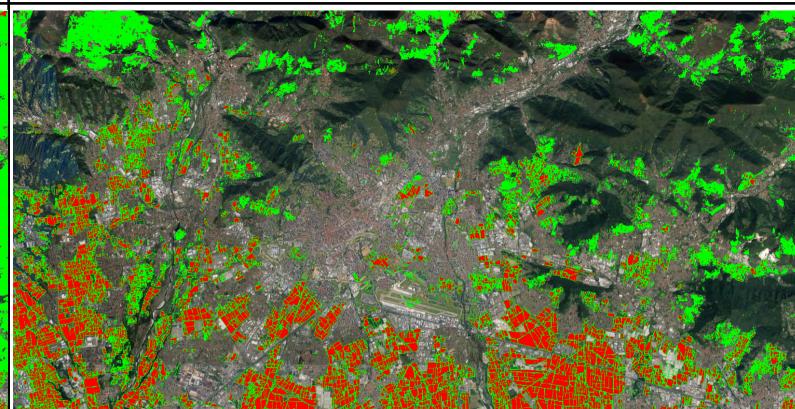


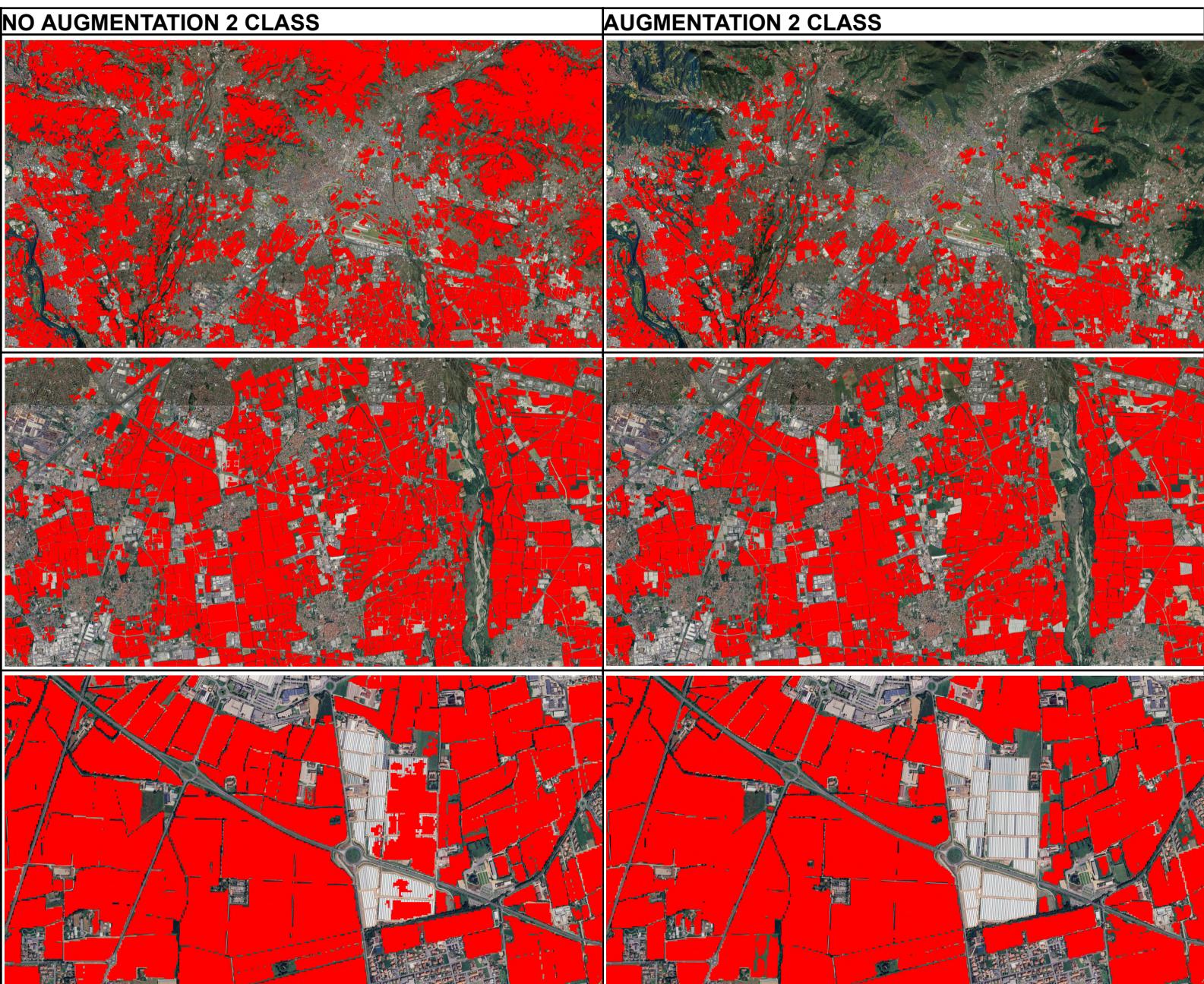
LE ETICHETTE 2 E 3 CLASSI NEI GRAFICI SONO INVERTITE

NO AUGMENTATION 3 CLASS



AUGMENTATION 3 CLASS





Risultati in accordo col paper, ma trovato un bug nel codice che non ha permesso di utilizzare data augmentation.

Gli esperimenti con data augmentation mostrano che la precisione è

8. Fase di sperimentazione

8.1. Modelli

8.1.1. UNet

8.1.1.1. BatchNorm

tutti BCE_DICE

MODEL	test_countries	pixel_level_iou	pixel_level_precision	pixel_level_recall	object_level_precision	object_level_recall	Trainable Parameters
UNET 2C (t52ta4g 1)	all (NO PO)	0.7829	0.8322	0.9296	0.1348	0.0759	31.0 M
UNET 2C BN (b6a7z5 4k)	all (NO PO)	0.8076	0.8625	0.9270	0.2707	0.1254	31.0 M

8.1.1.2. Parametri uguali a NoKAN

MODEL	test_countries	pixel_level_iou	pixel_level_precision	pixel_level_recall	object_level_precision	object_level_recall	Trainable Parameters
NOKAN 2C solo_lov asz (s9c33e5 g)	all (NO PO)	0.7876	0.8680	0.8947	0.4264	0.0904	2.8 M
UNET 2C solo_lov asz (2.8M) (cxxg2z5 8)	all (NO PO)	0.8066	0.8766	0.9099	0.3018	0.1507	2.8 M

8.1.1.3. Parametri uguali a KAN

tutti solo_lovasz

MODEL	test_countries	pixel_level_iou	pixel_level_precision	pixel_level_recall	object_level_precision	object_level_recall	Trainable Parameters
UKAN 2C (6.4M) (h852d9 7h)	all (NO PO)	0.7933	0.8658	0.9045	0.3989	0.1208	6.4 M

UNET 2C (6.4M) (dpqx99 dt)	all (NO PO)	0.8176	0.8826	0.9172	0.3355	0.1695	6.4 M
--	----------------	---------------	---------------	---------------	--------	---------------	-------

8.1.2. UKAN

8.1.2.1. Spline order

8.1.2.2. Grid Size

MODEL	test_cou ntries	pixel_lev el_iou	pixel_lev el_precis ion	pixel_lev el_recall	object_le vel_preci sion	object_le vel_recal l	Trainable Paramet ers
UKAN 2C solo_lov asz (6.4M) (h852d9 7h)	all (NO PO)	0.7933	0.8658	0.9045	0.3989	0.1208	6.4 M
UKAN 2C grid_size =15 spline_or ger=5 solo_lov asz (kwfqpn3 k)	all (NO PO)	0.7953	0.8699	0.9026	0.3983	0.1175	11.1 M
UKAN 3C grid_size =15 spline_or ger=5 BCE_Dic e (hvzva9 cc)	all (NO PO)	0.7372	0.8988	0.8038	0.3286	0.3037	11.1 M
UKAN 3C BCE_Dic e (vc3gob ch)	all (NO PO)	0.7308	0.8987	0.7964	0.3168	0.3000	6.4 M

8.1.2.3. Dimensione large

tutti solo_lovasz

MODEL	test_countries	pixel_leve_l_iou	pixel_leve_l_precision	pixel_leve_l_recall	object_leve_l_precision	object_leve_l_recall	Trainable Parameters
UKAN 2C solo_lovasz (6.4M) (h852d97h)	all (NO PO)	0.7933	0.8658	0.9045	0.3989	0.1208	6.4 M
UKAN large 2C solo_lovasz (jsw2zg wj)	all (NO PO)	0.8075	0.8790	0.9084	0.4224	0.1424	25.4 M

8.1.3. NoKAN

kufanxyh = pesi del modello uguali a quelli della unet, distribuiti solamente nella parte nokan
Parametri allenabili encoder1: 3552
Parametri allenabili encoder2: 14016
Parametri allenabili encoder3: 185088
Parametri allenabili block1: 807424
Parametri allenabili block2: 3187712

8.1.3.1. Dimensione UNet

MODEL	test_countries	pixel_leve_l_iou	pixel_leve_l_precision	pixel_leve_l_recall	object_leve_l_precision	object_leve_l_recall	Trainable Parameters
NOKAN 2C solo_lovasz (kufanxyh)	all (NO PO)	0.7909	0.8672	0.8998	0.4017	0.1219	27.7 M
unet_ori	all (NO)	0.8277	0.8904	0.9215	0.3731	0.1966	31.0 M

ginal 2C solo_lov asz (lnwt64z d)	PO)						
---	-----	--	--	--	--	--	--

8.1.3.2. Dimensione Ukan

MODEL	test_countries	pixel_level_iou	pixel_level_precision	pixel_level_recall	object_level_precision	object_level_recall	Trainable Parameters
NOKAN 2C solo_lov asz (6.4M)(z vpmzb0 4)	all (NO PO)	0.8013	0.8752	0.9046	0.4004	0.1318	6.4 M
ukan 2C solo_lov asz (h852d9 7h)	all (NO PO)	0.7933	0.8658	0.9045	0.3989	0.1208	6.4 M

AGGIUNTA OSSERVAZIONE SUL PARAGONE TRA LE 'ARCHITETTURE.

Nei paper in letteratura, la UKAN viene paragonata con la UNET, questo paragone è per noi non sensato. Per poter valutare se le KAN siano meglio bisognerebbe sostituire i layer standard con dei layer KAN, nel caso della UNET dovremmo sostituire i CONV con dei KANCONV, però questo non avviene per la UKAN, infatti vengono aggiunti dei KANLINEAR, cioè l'equivalente KAN di un MLPLAYER.

Quindi noi abbiamo confrontato la UKAN, con la NOKAN, cioè la medesima architettura dove al posto del KANLINEAR abbiamo sostituito un semplice LINEAR.

Qui si scopre che in realtà non ci sono particolari vantaggi nell'utilizzare i neuroni KAN.

8.1.4. Clay

AGGIUNGERE TABELLA CLAY OTTICO

Clay si è dimostrato valido, con prestazioni paragonabili ai migliori modelli ottenuti finora. Clay converge prima, questo grazie al suo pretraining, quindi bastano una ventina di epochhe soltanto. Potrebbe tornare utile il fatto che abbia un pretraining dato che per il DUSAf abbiamo pochi esempi con cui fare fine tuning. Inoltre il training con FTW comprende solamente zone di pianura, invece la lombardia comprende numerose montagne e si vede che in quelle zone fa male, clay d'altra parte possiede già una informazione globale che potrebbe migliorare le prestazioni lì dove FTW fallisce.

8.2. Loss

Una delle teorie di fondo alla base delle KAN, è il fatto che per un task più complicato potessero fare meglio.

Non potendo complicare il task è stata complicata la loss.

Sono state testate le seguenti loss:

MODEL	test_countries	pixel_level_iou	pixel_level_precision	pixel_level_recall	object_level_precision	object_level_recall	Trainable Parameters
ukan 3C BCE (pa5vhlf)	all (NO PO)	0.7819	0.8556	0.9007	0.3474	0.0942	6.4 M
ukan 3C bound_dice_steal_weights (f9ejl3u6)	all (NO PO)	0.7035	0.8657	0.7897	0.2964	0.0437	6.4 M
ukan 3C lovasz_boundary (0zmgas3w) idc = [1, 2], loss = 0.5 * bloss + 0.5 * lovasz	all (NO PO)	0.7105	0.8979	0.7729	0.4108	0.1769	6.4 M
ukan 3C bound_ce (833wkqpi) idc = [1, 2], loss = 0.5 * bloss + 0.5 * ce	all (NO PO)	0.7417	0.8891	0.8173	0.3651	0.2335	6.4 M

ukan 3C bound (1oeajbh7) idc [1, 2]	all (NO PO)	0.6670	0.8490	0.7568	0.2500	0.0306	6.4 M
ukan 3C solo_lovasz (ccbpgz32)	all (NO PO)	0.5361	0.9224	0.5614	0.1479	0.1169	6.4 M
UKAN 3C BCE_Dic e NOPESI silu (jjkji9nv)	all (NO PO)	0.7791	0.8564	0.8962	0.3388	0.0997	6.4 M
UKAN 3C BCE_Dic e (vc3gobch)	all (NO PO)	0.7308	0.8987	0.7964	0.3168	0.3000	6.4 M
ukan 2C bound_dice_stal_weights (11eyic19)	all (NO PO)	0.7032	0.8661	0.7889	0.3029	0.0431	6.4 M
ukan 2C bound_ce_stal_weights (lvyer29t)	all (NO PO)	0.7512	0.8779	0.8388	0.3950	0.0825	6.4 M
ukan 2C lovasz_boundary (7gyy6q6u) idc [1],	all (NO PO)	0.7286	0.8572	0.8291	0.3309	0.0539	6.4 M

loss = 0.5 * bloss + 0.5 * lovasz							
ukan 2C solo_lov asz (h852d9 7h)	all (NO PO)	0.7933	0.8658	0.9045	0.3989	0.1208	6.4 M
UKAN 2C BCE_Dic e (r1jxk36f)	all (NO PO)	0.7787	0.8356	0.9196	0.2917	0.0827	6.4 M
UKAN 2C BCE_Dic e NOPESI silu (retvtk7a)	all (NO PO)	0.7789	0.8671	0.8844	0.3521	0.1266	6.4 M

8.2.1. Dice

8.2.1.1. Pesata

8.2.1.2. Non pesata

MODEL	test_cou ntries	pixel_lev el_iou	pixel_lev el_precis ion	pixel_lev el_recall	object_le vel_preci sion	object_le vel_recal l	Trainable Paramet ers
UKAN 3C BCE_Dic e (vc3gob ch)	all (NO PO)	0.7308	0.8987	0.7964	0.3168	0.3000	6.4 M
UKAN 3C BCE_Dic e	all (NO PO)	0.7791	0.8564	0.8962	0.3388	0.0997	6.4 M

NOPESI silu (jjkji9nv)							
UKAN 2C BCE_Dic e (r1jxk36f)	all (NO PO)	0.7787	0.8356	0.9196	0.2917	0.0827	6.4 M
UKAN 2C BCE_Dic e NOPESI silu (retvtk7a)	all (NO PO)	0.7789	0.8671	0.8844	0.3521	0.1266	6.4 M

8.2.2. Lovaz

8.2.3. Boundary loss

8.3. Dati

8.3.1. Sentinel 1

Il dataset di sentinel 1 è una replica approssimata del dataset FTW ma con immagini radar.

Le differente:

Per quanto riguarda le acquisizioni, non essendo possibile avere una acquisizione nello stesso identico momento, si è proceduto nel seguente modo:

- per ogni macro griglia di FTW, sono state identificate le fasce temporali per ogni paese in cui i campi avessero la maggiore varianza (vedi tabella nel loro paper) ed è stata presa la prima acquisizione valida che la coprisse per intero. Questo è stato fatto sia per la finestra temporale a che per la b. L'acquisizione copre l'intera macro griglia.

C'è da considerare che il dataset così ottenuto è più piccolo, dato che per alcune macro griglie non sono state trovate acquisizioni valide, cioè che non rispettassero i requisiti della query.

MODEL	test_cou ntries	pixel_lev el_iou	pixel_lev el_precis ion	pixel_lev el_recall	object_le vel_preci sion	object_le vel_recal l	Trainable Paramet ers
UNET 2C solo_lov asz SAR (vvlqk68 8)	all (NO PO)						31.0 M
UKAN	all (NO						6.4 M

2C solo_lov asz SAR (vlfin8ou)	PO)						
NOKAN 2C solo_lov asz SAR (svgxf6zx)	all (NO PO)						2.8 M
CLAY unet_he ad 2C solo_lov asz SAR (1vourdx 5)	all (NO PO)						43.1 M (354 M total)

8.3.2. Sentinel 2

Il dataset di sentinel 2 è esattamente quello proposto dal paper FTW

8.4. Predizione

8.4.1. Due classi

8.4.2. Tre classi

8.5. Metriche

spiegare quali metriche abbiamo usato e perchè

8.5.1. Object metrics

8.5.2. IoU

9. Risultati

9.1. Sentinel 1

9.1.1.

9.2. Sentinel 2

9.2.1. Unet vs Nokan vs Dice

Osservazioni

-Le reti non apprendono direttamente il concetto di poligono, pertanto ci si ritrova in situazioni in cui si formano irregolarità dentro i bordi dell'immagine. Al tempo stesso capita che bordi tra loro vicini all'adiacenza non siano ben definiti.

-La segmentazione prodotta da FOTW non è ottenuta interamente mediante ispezione visiva delle immagini date in input ai modelli, questo può generare introduzione di rumore nel processo di apprendimento: se un poligono è segnato come campo per rilevazioni svolte sul posto ma non è possibile inferire che tale segmento di immagine risulti un campo ciò può ridurre le performance a causa delle fluttuazioni che avvengono durante il processo di apprendimento.

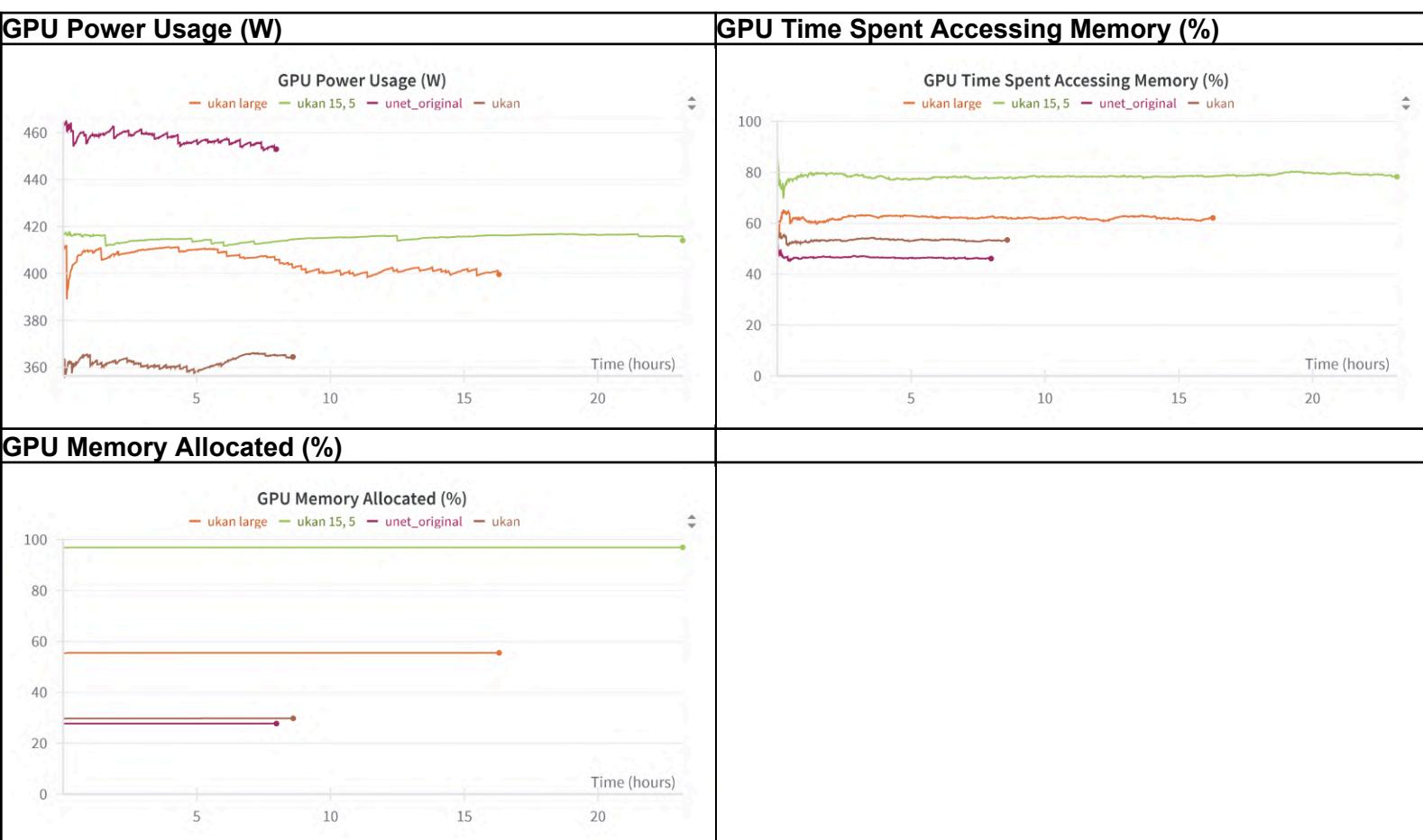
-Non si notano differenze di performance significative tra due classi e tre classi rispetto ad IoU, diversa è la situazione della metrica Object che pare giovare dall'introduzione della classe bordo.

-La UKAN si dimostra decisamente inefficiente riguardo il tempo speso in lettura della VRAM, rendendo il modello più lento in fase di addestramento e facendo diminuire il quantitativo di potenza media impiegata. Ciò non si traduce in un vantaggio, **è da verificare se nel paper della Baralis il valore immesso della potenza richiesta è ridotto a causa di questo problema, se così fosse un fatto ancor più rilevante dal momento che correggerebbe ambiguità presenti in letteratura.**

Dalle seguenti tabelle si vede infatti che si ha una differenza di 100W.

Il tempo di accesso alla memoria VRAM è più alto, specialmente con l'aumentare del grid size e dello spline order.

Il costo in VRAM della GPU è molto più alto per le KAN.



- Sebbene ci fossero aspettative riguardo la complicazione della loss a vantaggio della UKAN specialmente con l'aumento di spline order e grid size, non si sono apprezzati miglioramenti degni di nota

-Ukan e Nokan performano meglio qualora la maschera sia ruotata rispetto all'immagine da segmentare di Unet, probabilmente a causa delle trasformazioni non locali imputabili al fatto che non usano interamente convoluzioni come accade invece per Unet.

-La segmentazione prodotta dalla tesista da utilizzare come ground truth non è di ottima qualità

