Fashion-MNIST Image Classification

Paolo Leopardi

***Abstract*—Image Classification è il processo che consente di classificare un'immagine rispetto al contenuto informativo della stessa,assegnandole quindi un qualche significato semantico. Il dataset utilizzato per lo sviluppo del progetto è Fahion-MNIST, questo è costituito da immagini di indumenti di vario tipo. Viene proposto un approccio tramite tecniche di deep learning utilizzando due differenti tipi di reti neurali convoluzionali: il primo modello è stato implementato da zero, il secondo è invece un'architettura di riferimento nel campo delle reti convoluzionali. I risultati mostrano come un modello convoluzionale piuttosto semplice riesca ad ottenere dei risultati molto soddisfacenti evidenziando ancora una volta la potenza delle reti neurali.**

I. Introduction

Il task di Image Classification consiste nell'assegnare ad un'immagine una specifica etichetta scelta fra un set di label disponibili. Le immagini di cui si compone il dataset dovranno raffigurare perciò un solo elemento fra quelli associabili ad una delle classi. A differenza di altri task tipici della Computer Vision questo non richiede alcuna localizzazione degli elementi presenti nell'immagine; basti pensare ad esempio all'Object Detection and Recognition in cui, oltre alla classificazione dell'oggetto presente nell'immagine, si devono fornire anche le coordinate della boundig box che lo racchiude.

In questo progetto è stato impiegato il dataset Fashion-MNIST [1], liberamente scaricabile da GitHub. Quest'ultimo è costituito da immagini grayscale 28x28 di capi d'abbigliamento provenienti dal sito di e-commerce Zalando (Figura 1) classificati in classi da 0 a 9 nel seguente ordine: T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot. La nascita di Fashion-MNIST è dovuta al sovrautilizzo del noto dataset MNIST database of handwritten digit [2], utilizzato come punto di riferimento dalla comunità scientifica. Gli autori infatti spiegano che MNIST handrwitten è un dataset "troppo facile" dato che le reti convoluzionali riescono a raggiungere un punteggio del 99,7% ed anche gli algoritmi classici di Machine Learning riesco a ottenere facilmente score del 97%. L'esperto di deep learning François Chollet, creatore di Keras [3], ha dichiarato nell'Aprile del 2017 attraverso il proprio profilo Twitter che MNIST non è rappresentativo dei moderni task di CV.

Per il task in questione sono state utilizzate due soluzioni architetturali differenti di reti neurali convoluzionali, le reti di questo tipo sono infatti disegnate per processare i dati disposti a griglia (come accade per i pixel di un'immagine). Una rete convoluzionale è un caso speciale di rete neurale che utilizza la convoluzione in almeno uno dei suoi layer. Le architetture proposte sono:

- CustomNet
- ResNet-18 [4]

CustomNet è un'architettura molto semplice progettata da zero ed è costituita da 3 layer convoluzionali e da un ultimo layer fully connected. ResNet-18 è invece il modello proposto da He et al. opportunamente riadattato per lo specifico compito in questione.

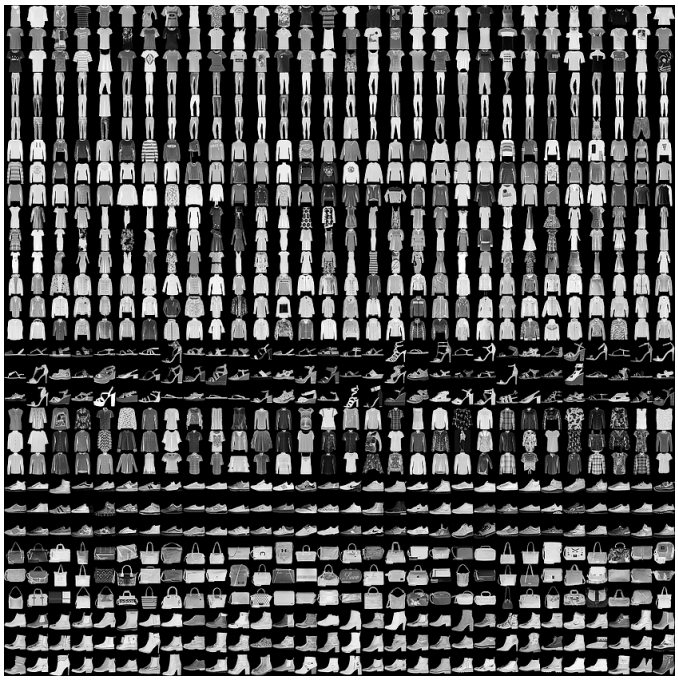

Fig. 1: Immagini estratte dal dataset Fashion-MNIST (tre righe per classe)

Infine sono state testate le soluzioni architetturali proposte valutandone le performance grazie alle metriche di accuracy, precision e recall.

II. Related Work

Ad oggi esistono varie architetture convoluzionali fondamentali che sono divenute famose nel corso degli anni, ognuna avente le sue peculiarità; molte di queste sono state utilizzate per task di Image Classification: LeNet-5 [5], AlexNet [6], VGGNet[7]. Le reti sopracitate insieme ad altre sono state valutate anche in Fashion-MNIST da Y.Zhang in un articolo pubblicato nel 2019 [8].

Oltre alle architetture fondamentali vengono proposti innumerevoli pattern che riescono ad ottenere degli ottimi risultati. Ad esempio il modello sviluppato da Duan et al. [9] basato su VGG ha raggiunto un'accuracy del 91,5%. Un'altra soluzione realizzata da Bhatnagar et al. [10] ha ottenuto il 92,54%.

III. Proposed Approach

Il dataset prevede 60.000 campioni di training e 10.000 di test, è stato però necessario ridurre il numero di sample per il training set così da avere delle tempistiche ragionevoli

per effettuare l'addestramento delle reti. Sono quindi stati estratti i primi 5.000 sample per il training e di conseguenza la dimensione del test set è stata portata a 1.000 campioni. Come mostrato nelle Figure 2 e 3 la distribuzione delle etichette rimane comunque bilanciata sia nel training che nel test set.

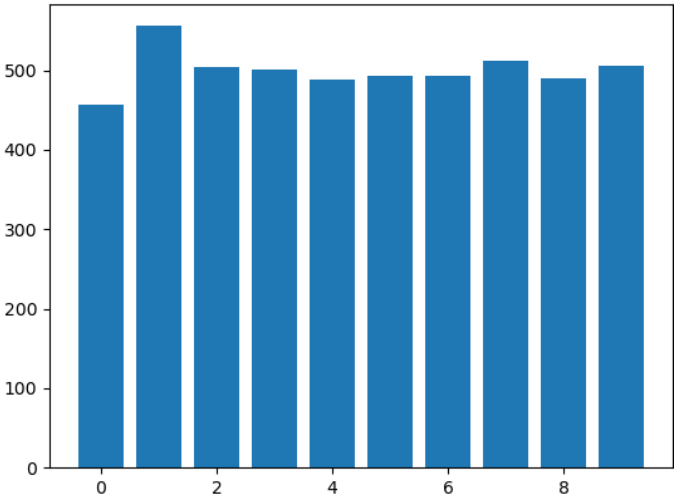

Fig. 2: Distribuzione label training set dopo la riduzione del dataset

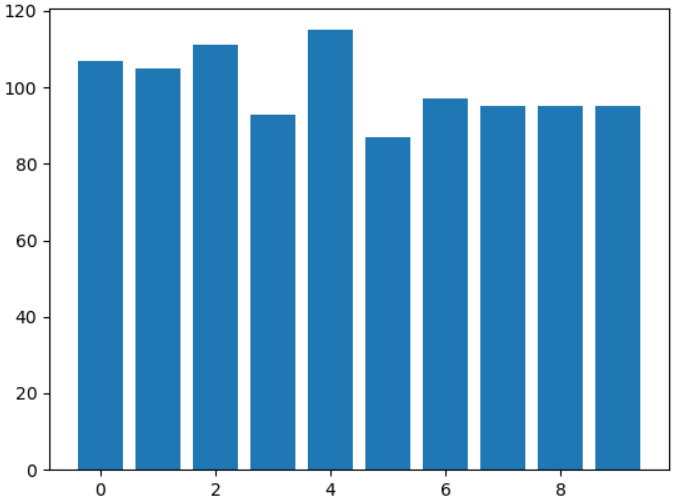

Fig. 3: Distribuzione label test set dopo la riduzione del dataset

CustomNet

L'architettura (Figura 4) è costituita da tre blocchi convoluzionali, i primi due del tutto identici fra loro, il terzo è stato leggermente modificato rispetto ai precedenti:

1) **Convolutiona layer**
   input ch: 1
   out ch: 32
   kernel size: 3×3
   stride: 1
   padding: 1
2) **Batch Normalization**
3) **ReLU**
4) **Max Pooling 2D**
   kernel size: 2×2
   stride: 2
5) **Convolutional layer**
   input ch: 32
   out ch: 64
   kernel size: 3×3
   stride: 1
   padding: 1
6) **Batch Normalization**
7) **ReLU**
8) **Max Pooling 2D**
   kernel size: 2×2
   stride: 2
9) **Convolutional layer**
   input ch: 64
   out ch: 128
   kernel size: 3×3
   stride: 1
   padding: 1
10) **Batch Normalization**
11) **ReLU**
12) **Dropout**
    probability: 0.5
13) **Fully connected**
    in size: 6272
    out size: 10

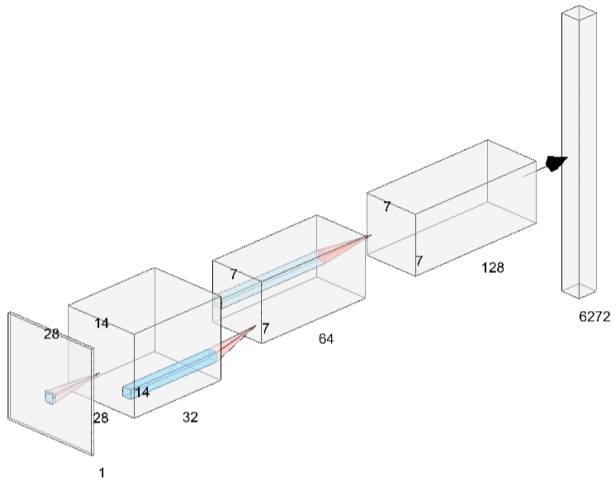

Fig. 4: Architettura CustomNet

In tutti i blocchi la dimensione del kernel è 3×3, impilare layer convoluzionali con kernel di dimensioni ridotte consente di avere una maggior espressività delle feature ed un minor numero di parametri rispetto ad un unico layer convoluzionale avente un kernel di dimensioni maggiori. È stato sempre applicato il padding same così da non ridurre la dimensione del tensore in input al blocco successivo a seguito della convoluzione. Successivamente è stato inserito il layer di batch normalization così da normalizzare l'output della convoluzione che poi andrà in input alla funzione di attivazione. ReLU (Rectified Linear Unit) è una funzione non lineare definita

come:

$$f(x) = max(0, x)$$

Nei primi due blocchi è stato applicato un max pooling, questo consente di ottenere una statistica sommaria degli output del layer precedente nell'intorno del pixel e una rappresentazione invariante a piccole traslazioni dell'input. I parametri impostati per il pooling dimezzano la dimensione del volume in ingresso. In Figura 4 è mostrato un esempio di applicazione di max pooling. Nell'ultimo blocco al posto del max pooling è stato

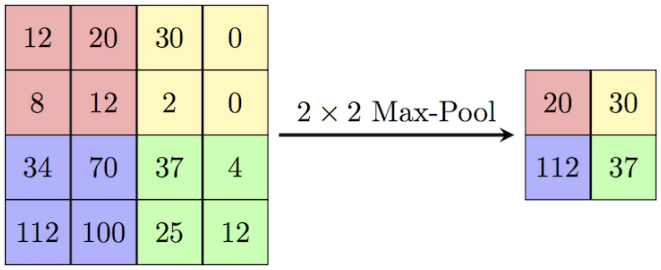

Fig. 5: Applicazione max pooling con kernel 2×2 e stride 2

introdotto un layer di dropout [11] per effettuare la regolarizzazione e quindi prevenire l'overfitting.Il dropout è una tecnica di regolarizzazione estremamente efficace che consiste nel mantenere un neurone attivo con probabilità p, questo si traduce in fase di training con l'eliminazione di alcune connessioni randomicamente in base a p. Così facendo la rete utilizza un numero di parametri inferiore e non sapendo a priori quali connessioni verranno inibite è costretta ad apprendere i concetti veramente importanti. Infine vi è il layer fully connected che produce in uscita la predizione, chiaramente è stato necessario effettuare precedentemente la vettorializzazione del tensore.

Resnet-18

Essendo Resnet-18 un'architettura fondamentale non è stato necessario implementarla da zero, tuttavia la struttura è stata modificata in alcuni punti per adattarla al task in questione. È stato sostituito il primo layer convoluzionale dato che questo prende in input immagini RGB, quindi aventi 3 canali. Il nuovo layer, sempre convoluzionale, ha gli stessi parametri dell'originale fatta eccezione per la dimensione dell'input che passa da 3 a 1. A valle della rete l'ultimo layer fully connected è stato modificato per avere l'uscita di dimensione pari al numero delle classi, cioè 10.

È stata scelta questa rete essendo uno standard attuale che rappresenta la backbone delle moderne reti neurali, è stato inoltre il progetto vincitore nel 2015 di ILSVRC [12] (ImageNet Large Scale Visual Recognition Challenge).

L'elemento caratterizzante di questa rete sono i residual block: questi consento al gradiente di propagarsi all'indietro nella rete senza essere alterato cosi da scongiurare il problema del *vanishing gradient*.

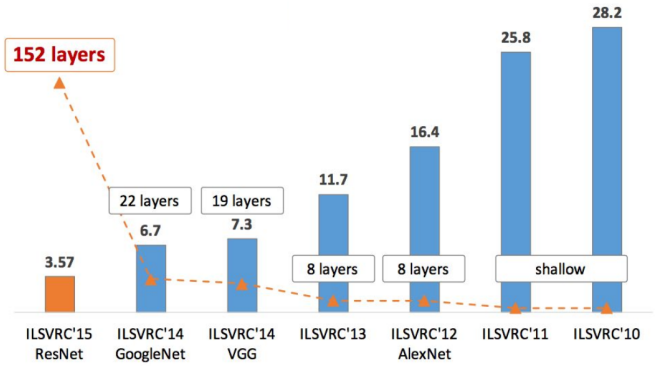

Fig. 6: Vincitori di ILSVRC dal 2010 al 2015

IV. Experiments

Training

Entrambe le reti sono state addestrate per 30 epoche utilizzando la cross-entropy loss. CustomNet è stata addestrata da zero e la Figura 7 mostra i filtri del primo layer convoluzionale al termine dell'addestramento.

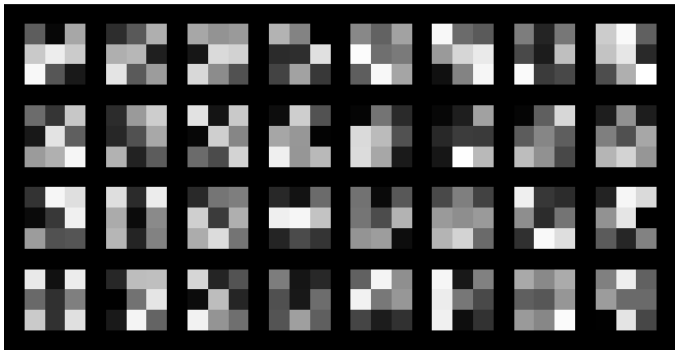

Fig. 7: Filtri del primo layer convoluzionale di CustomNet ottenuti al termine della fase di training

La figura 8 mostra la decima feature map per ogni layer convoluzionale; la prima immagine mostra l'input, in questo caso corrispondente alla classe Coat (4). Si osserva come man mano che si scende in profondità la rete impara ad estrarre l'oggetto presente e distinguerlo dal background.

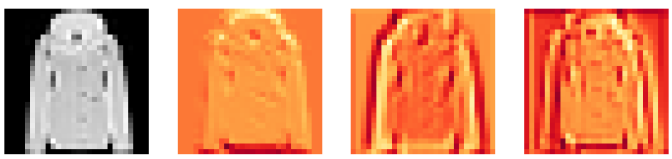

Fig. 8: Input e feature map dei layer convoluzionali

Per quanto riguarda Resnet sono stati addestrati solo i layer alterati rispetto alla struttura originale, essendo la rete preaddestrata con il dataset ImageNet [13]. Grazie a questa procedura di transfer learning è stato possibile utilizzare la rete che altrimenti avrebbe richiesto un tempo eccessivo per la fase di training.

Risultati

Per valutare le performance delle reti è stata utilizzata l'accuracy, questa consente di individuare la percentuale di classificazioni corrette sul totale delle predizioni. È definita come:

$$Accuracy = \frac{predizioni\ corrette}{predizioni\ totali}$$

Le Figure 9 e 10 mostrano l'andamento dell'accuracy su train e test set per entrambi i modelli all'avanzare delle epoche. CustomNet riesce a raggiungere un'accuracy del 91% sul train set e dell'88% sul test. Resnet-18 ottiene dei risultati inferiori arrivando ad un 74% sul training e un 68% sul test.

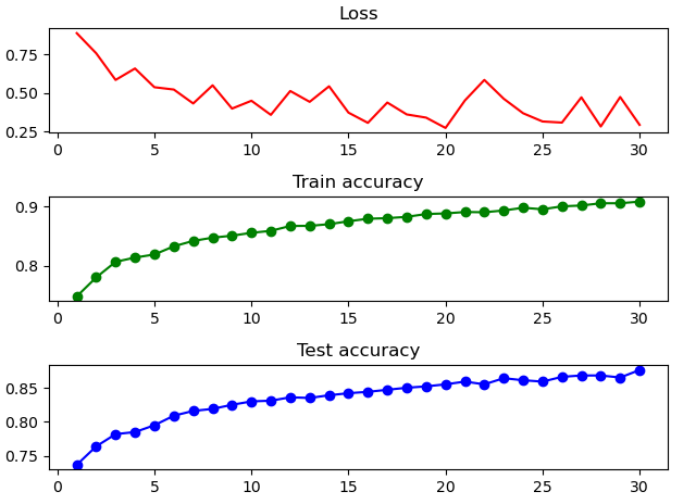

Fig. 9: Andamento di loss, train accuracy e test accuracy utilizzando CustomNet

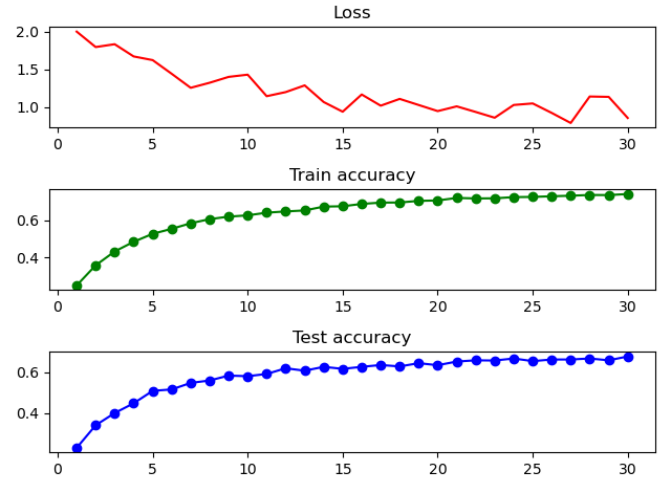

Fig. 10: Andamento di loss, train accuracy e test accuracy utilizzando ResNet-18

Sono state inoltre calcolate precision e recall per le singole classi:

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

La precision indica la proporzione di predizioni positive corrette rispetto a tutte le predizioni positive. La recall indica invece la proporzione di predizioni positive corrette rispetto a tutti i campioni effettivamente positivi. In un task di classificazione multiclasse si assegna la classe positiva ad una sola etichetta e al resto delle label viene assegnata la classe negativa.

	CustomNet		**ResNet-18**	
Label	**Precision**	**Recall**	**Precision**	**Recall**
0: T-shirt/Top	0.85	0.90	0.67	0.73
1: Trouser	0.98	0.99	0.90	0.81
2: Pullover	0.82	0.81	0.54	0.43
3: Dress	0.84	0.85	0.67	0.64
4: Coat	0.83	0.86	0.52	0.65
5: Sandal	0.89	0.94	0.76	0.80
6: Shirt	0.76	0.68	0.34	0.37
7: Sneaker	0.92	0.90	0.85	0.84
8: Bag	0.95	0.96	0.75	0.79
9: Ankle boot	0.95	0.92	0.82	0.82

TABLE I: Precision e recall calcolate sul test set

Infine è stata visualizzata la confusion matrix per CustomNet riguardante le predizioni effettuate sul test set. Grazie a questa si riesce a capire quali classi siano più difficili da disambiguare per la rete. La diagonale principale rappresenta il numero di predizioni corrette per ogni classe, mentre invece i valori fuori diagonale rappresentano una predizione sbagliata.

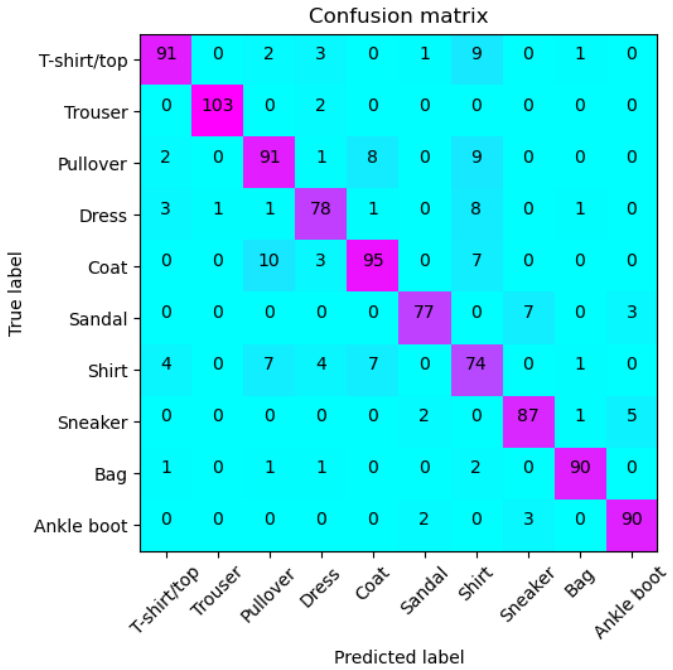

Fig. 11: Confusion matrix per le predizioni di CustomNet sul test set

V. Conclusion

I risultati ottenuti evidenziano come CustomNet seppur essendo una rete piuttosto semplice riesca comunque a realizzare dei buoni score. Riguardo ResNet invece i risultati

sono significativamente inferiori, tuttavia bisogna tener sempre in considerazione che la rete è stata scaricata preaddestrata su immagini di tipo RGB provenienti da un altro dataset; solamente il primo e l'ultimo layer sono stati addestrati sul training set di Fahion-MNIST.

Entrambe le reti mostrano comunque una forte insofferenza rispetto alla classe numero 6, ovvero Shirt. Come si può vedere dalla confusion matrix questa viene confusa da CustomNet con le classi T-shirt/top, Pullover, Dress e Coat. La figura 12 mostra un esempio di queste classi e si intuisce come ad esempio le classi Shirt e T-shirt/top possano risultare molto simili. La rete riesce invece a distinguere bene le

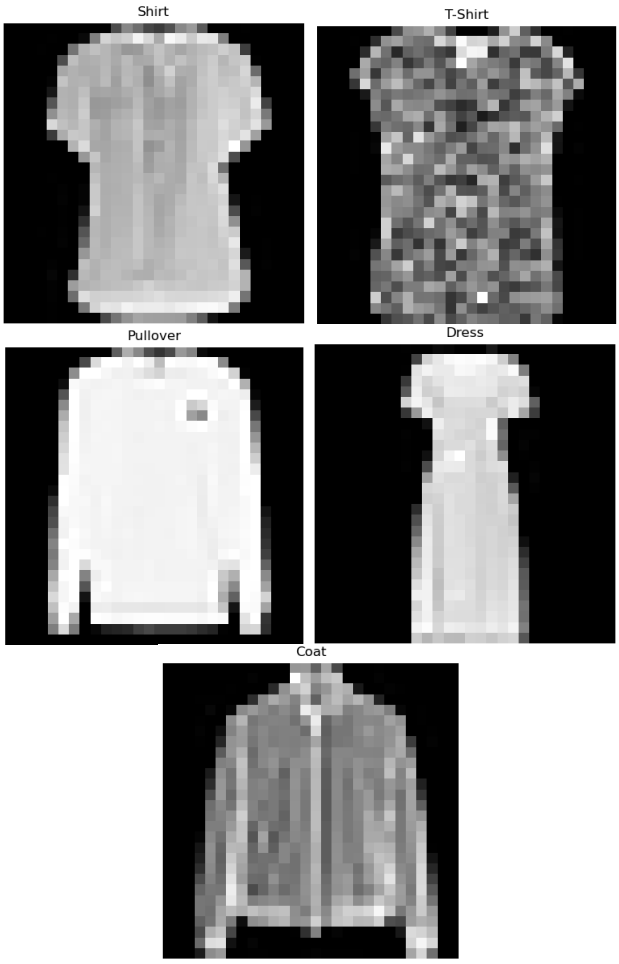

Fig. 12: Esempi di immagini per le classi Shirt, T-shirt/top, Pullover, Dress e Coat

classi riguardanti le calzature (Sandal, Sneaker, Ankle boot) seppur possano sembrare analoghe fra loro in alcuni casi. Le immagini raffiguranti le label Trouser e Bag vengono classificate sempre in maniera corretta fondamentalmente. I due indumenti hanno infatti delle forme molto differenti rispetto alle altre classi, basta infatti osservare la Figura 1 a monte del documento per comprenderne le differenze.

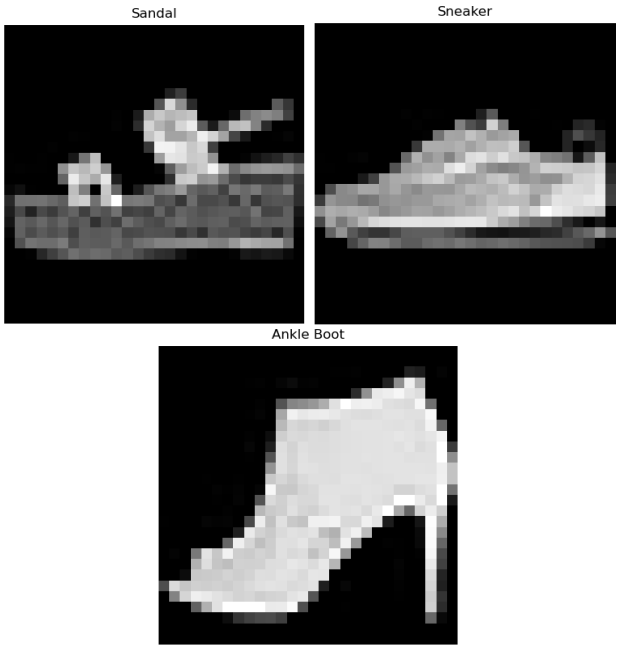

Fig. 13: Esempi di immagini per le classi Sandal, Sneaker, Ankle boot

REFERENCES

[1] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[2] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[3] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[8] Y. Zhang, "Evaluation of cnn models with fashion mnist data," 2019.

[9] C. Duan, P. Yin, Y. Zhi, and X. Li, "Image classification of fashion-mnist data set based on vgg network," in *Proceedings of 2019 2nd International Conference on Information Science and Electronic Technology (ISET 2019). International Informatization and Engineering Associations: Computer Science and Electronic Technology International Society*, vol. 19, 2019.

[10] S. Bhatnagar, D. Ghosal, and M. H. Kolekar, "Classification of fashion article images using convolutional neural networks," in *2017 Fourth International Conference on Image Information Processing (ICIIP)*. IEEE, 2017, pp. 1–6.

[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.