



Università
di Catania

DIPARTIMENTO DI MATEMATICA E INFORMATICA
CORSO DI LAUREA MAGISTRALE IN INFORMATICA

Raffaele Terracino

Rimozione di artefatti in un dataset di mammografie

RELAZIONE PROGETTO DI MULTIMEDIA

Prof: Dario Allegra
Prof: Filippo Stanco

Anno Accademico 2024 - 2025

Indice

| | |
|--|-----------|
| Introduzione | 1 |
| Metodi | 3 |
| Ribaltamento orizzontale | 4 |
| Rimozione del testo | 5 |
| Post processing | 7 |
| Rimozione di piccole regioni bianche | 10 |
| Risultati | 14 |
| Conclusione | 17 |

Introduzione

Il progetto di Multimedia svolto consiste nella rimozione di artefatti in un dataset di immagini di mammografie. Il dataset utilizzato è il "MIAS Mammography" dataset, consistente in 322 immagini di 1024x1024 pixel in scala di grigio, raffiguranti mammografie effettuate con lo scopo di diagnosticare la presenza di "anomalie", presumibilmente tumori, presenti nella regione mammaria. Il dataset "MIAS Mammography" è distribuito secondo una licenza che ne consente l'uso per "scopi di ricerca" e che richiede di citare gli autori del dataset. Per il suddetto progetto, i requisiti della licenza vengono rispettati e sono racchiusi nel file "README.txt" della cartella di progetto. Le immagini sono memorizzate in formato PGM e sono numerate da 1 a 322 con prefisso "mdb". Ogni coppia di immagini raffigura entrambi i seni di uno stesso soggetto: per esempio, l'immagine di numero 1 è la mammografia del seno sinistro, mentre l'immagine successiva, la numero 2, è quella del seno destro. Oltre alle immagini, sono presenti due file denominati "Info.txt". Il primo, presente all'interno della cartella contenente le immagini, è descrittivo del dataset e della licenza con cui viene distribuito, rappresentando quindi l'insieme di metadati associati al dataset. Il secondo invece è un dataset tabellare che per ogni immagine descrive le caratteristiche dell'anomalia riscontrata, se presente. In totale, le mammografie in cui sono state rilevate anomalie sono 119. Le modeste dimensioni del dataset hanno consentito di visualizzare le imma-

gini per verificare la presenza di artefatti. Così facendo, si è scoperto che 221 immagini presentano, accanto al seno, dei blocchi di testo identificativi del dispositivo di acquisizione. Questi blocchi di testo sono composti da caratteri neri su sfondo bianco. Oltre questo, alcune immagini presentano delle strisce di pixel bianchi, oblique od orizzontali, dovute a qualche tipo di movimento involontario del dispositivo o del soggetto durante l'acquisizione. L'obiettivo del progetto è di rimuovere questi artefatti, utilizzando le tecniche di image processing viste a lezione, preparando così le immagini per altri tipi di analisi, per esempio la costruzione di un classificatore. Il progetto è interamente sviluppato come notebook jupyter, usando il linguaggio Python.

Metodi

La pipeline di rimozione degli artefatti si articola di 3 fasi principali: il ribaltamento orizzontale dei seni sinistri, la rimozione dei blocchi di testo e la rimozione di strisce bianche. Si effettua una distinzione tra le strisce bianche piccole, che spaziano su al più 100 colonne, da quelle grandi, che spaziano su più di 100 colonne. Per ogni sezione, vengono presentati degli esempi con due immagini che spiegano le operazioni principali effettuate dagli algoritmi.

Ribaltamento orizzontale

Per rendere più omogeneo il dataset e per facilitare le successive operazioni, viene stabilita la convenzione che tutte le immagini debbano essere rivolte verso destra. Poichè, per come è costruito il dataset, tutte le immagini di numero dispari sono rivolte verso sinistra, basta aprirle ed utilizzare il metodo `cv2.flip` della libreria OpenCV passando come argomento un intero, pari a 1, che rappresenta il ribaltamento orizzontale. Per ogni immagine ribaltata vengono anche modificati i dati relativi all'anomalia, se presente. Poichè il ribaltamento è orizzontale e, come descritto nei metadati, il sistema di riferimento utilizzato nel dataset ha l'origine nel pixel in basso a sinistra, la coordinata x dell'anomalia viene sostituita con il valore $1024 - x$.

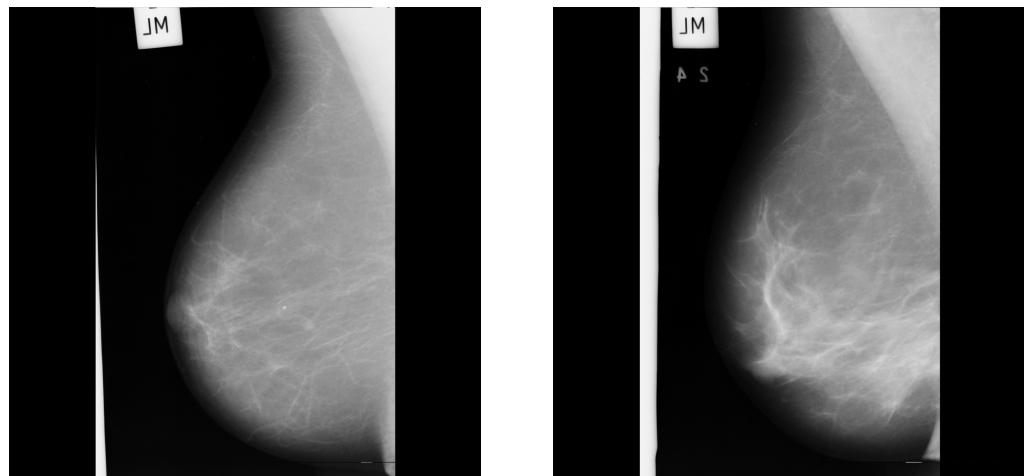


Figura 1: Immagini di input

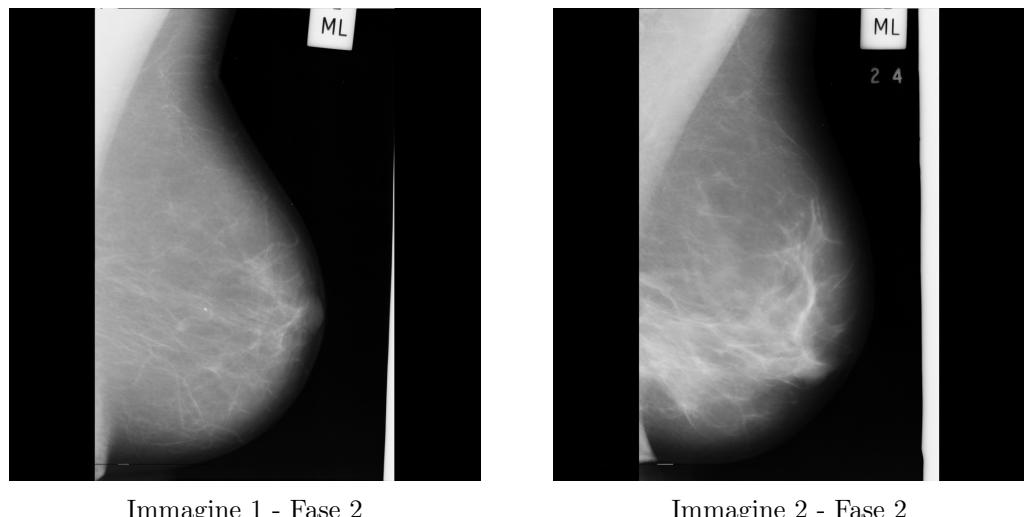


Immagine 1 - Fase 2

Immagine 2 - Fase 2

Figura 2: Immagini ribaltate

Rimozione del testo

L'algoritmo di rimozione del testo si articola in più fasi e fa uso di una combinazione di filtraggio spaziale, morfologia matematica ed edge detection per identificare le regioni da rimuovere e sostituirle con pixel neri. L'algoritmo è

implementato nella funzione removeTexts che prende in input un’immagine rappresentata come array bidimensionale, passata per riferimento. Il primo passo è effettuare una copia dell’immagine, su cui verranno eseguite tutte le operazioni. Di questa copia viene effettuato un cropping, prendendo soltanto le prime 400 righe e le colonne dalla 400 in poi. Questa operazione è fondamentale, in quanto restringere l’area di rilevazione degli artefatti è necessario per cercare di evitare regioni prive di artefatti. I valori scelti per il cropping, così come degli altri algoritmi utilizzati, sono empirici e sono stati scelti in base a vari test effettuati. Tali valori non sono ottimali per tutte le 211 immagini, perchè, sebbene gli artefatti siano principalmente presenti nella zona in alto a sinistra dell’immagine, nella zona di cropping potrebbero esserci anche parti di seno che vengono erroneamente considerate. Il secondo step è l’applicazione di un filtro gaussiano di dimensione 63, per ridurre il rumore nella regione croppata. Facendo la differenza tra la l’immagine croppata e il risultato del filtraggio si ottiene un’immagine in cui gli artefatti sono enfatizzati. Successivamente, su tale immagine differenza viene applicata una apertura, con kernel ellittico di dimensione 3, per eliminare piccoli regioni bianche che potrebbero compromettere i passi successivi. Il passo successivo è l’applicazione dell’algoritmo di Canny, utilizzando un kernel del LoG di dimensione 3 e piccoli valori per la sogliatura. Utilizzando un kernel ellittico, le regioni identificate vengono dilatate. L’obiettivo adesso è la creazione di una maschera binaria che identifichi i contorni, compresa la regione interna da essi identificati. Per costruire la regione interna, si utilizza il metodo drawContours di openCV, migliorando il risultato costruendo un involucro convesso. A questo punto, si costruisce la maschera finale complementando l’involucro convesso e applicando una apertura per eliminare piccole regioni residue. L’immagine finale è ottenuta facendo l’and logico tra la regione croppata iniziale e sé stessa, uti-

lizzando la maschera binaria creata. Di seguito vengono illustrate le maschere create dall'algoritmo per le immagini in Figura 1 e l'output dell'algoritmo.



Immagine 1 - Fase 3

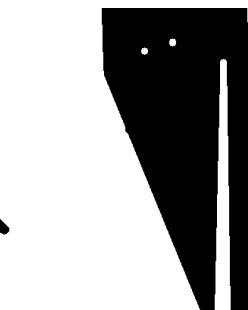


Immagine 2 - Fase 3

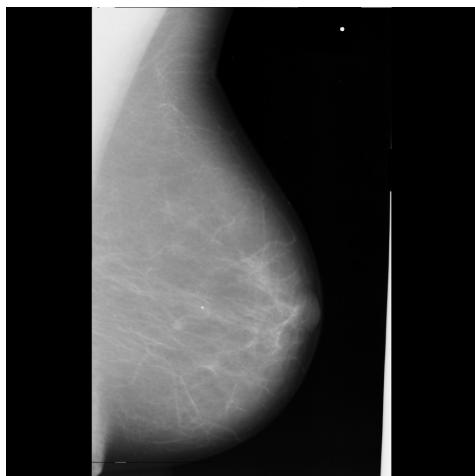
Figura 3: Maschere create dall'algoritmo di rimozione del testo

Immagine 1 - Fase 4

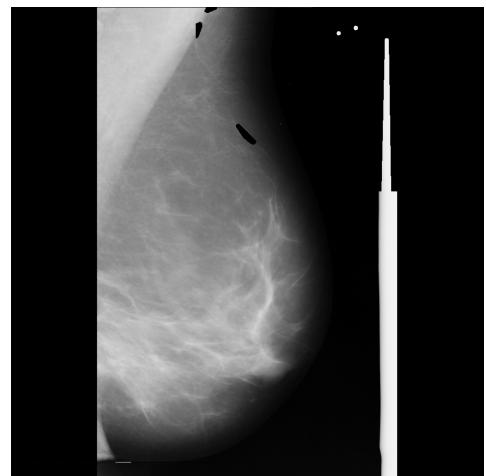


Immagine 2 - Fase 4

Figura 4: Output della rimozione del testo. Si notino gli artefatti prodotti.

Post processing

L'algoritmo proposto per la rimozione degli artefatti è stato pensato per essere eseguito su tutte le 221 immagini da correggere. Pertanto, tutti i parametri

degli algoritmi utilizzati al suo interno sono stati scelti in modo tale da avere buoni risultati su tutte le immagini. Tuttavia, a causa di questo, per alcune immagini l'algoritmo produce delle piccole regioni, approssimativamente circolari, di pixel neri o bianchi, che possono essere dovute alla scelta della regione di cropping o alle sogliature impostate per l'edge detection con Canny. Queste problematiche vengono risolte con una fase di post processing, effettuata come segue. Dapprima viene effettuato un cropping dell'immagine output dell'algoritmo, prendendo le prime 512 righe e le colonne a partire dalla 320. Successivamente vi è l'applicazione di Canny, seguita dal riempimento dei contorni e dalla costruzione di un involucro convesso, similmente all'algoritmo precedente. Così facendo, vengono identificati i buchi creati dal precedente algoritmo. A questo punto, bisogna distinguere le regioni in cui mettere 0 da quelle in cui sostituire il colore dell'immagine di partenza. Viene impostato 0 se, nella posizione individuata, il numero di colonna è maggiore di 200 e nell'immagine originale quella posizione ha valore di grigio maggiore di 220. Queste due condizioni permettono di distinguere i buchi della regione del seno da quelli presenti nella regione del blocco di testo da rimuovere. Di seguito vengono illustrate le maschere prodotte, identificative degli errori commessi dall'algoritmo precedente, e l'output della fase di post processing.



Figura 5: Maschere prodotte nella fase di post processing

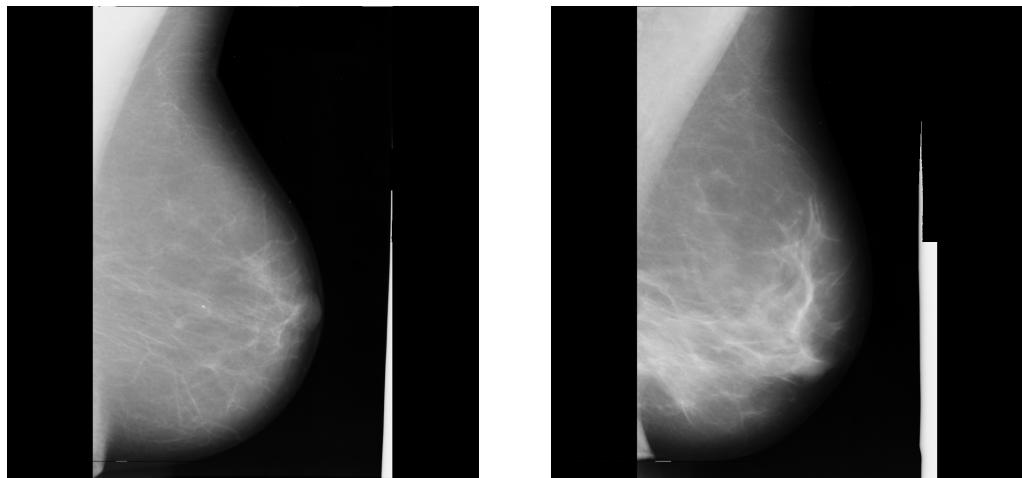


Figura 6: Correzione delle immagini mediante il post processing

Rimozione di grandi strisce bianche

Tra le 211 immagini da correggere, sono state identificate 49 immagini che presentano delle strisce di pixel bianchi di grandi dimensioni. Le strisce presenti sono per la maggior parte oblique. Tra queste, 38 richiedono anche la rimozione dei blocchi di testo con le procedure precedenti. Il numero esiguo di immagini ha permesso l'identificazione manuale delle regioni da correggere. Le regioni di interesse vengono corrette mediante utilizzo di un file csv di supporto, dove è indicato il numero di colonna dal quale in poi impostare i valori di grigio dei pixel a 0. Di seguito vengono illustrate le immagini in cui è stato applicato questo procedimento, che non necessitano più di ulteriore correzioni. Per le due immagini della Figura 1 si può dire che la pipeline sia giunta al termine.

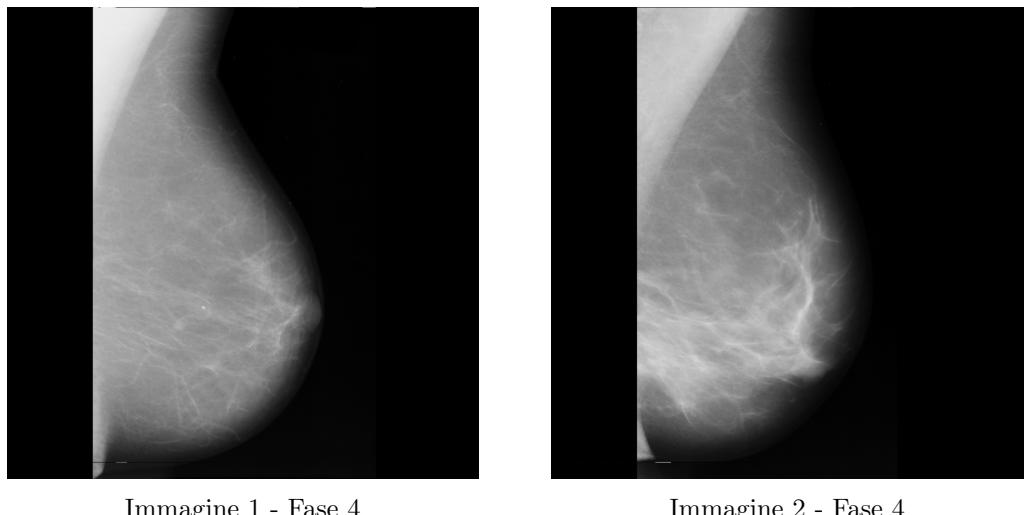


Immagine 1 - Fase 4

Immagine 2 - Fase 4

Figura 7: Rimozione delle strisce bianche e conseguente output finale della pipeline

Rimozione di piccole regioni bianche

Per migliorare ulteriormente il processo, per un sottoinsieme delle immagini si effettua anche la seguente procedura di eliminazione di piccole regioni bianche, che possono essere strisce o piccole forme geometriche. Della parte di immagine in cui tali artefatti sono presenti, si effettua un miglioramento del contrasto, seguito dall'applicazione di una sogliatura binaria di valore 170. Sul risultato viene effettuata un'apertura con un'elemento strutturante ellittico piccolo. Il complemento della differenza tra l'apertura e l'immagine sogliata identifica le regioni da correggere, i cui pixel assumono il colore nero. Di seguito viene illustrato il procedimento per due immagini che presentano piccole regioni bianche, mostrando anche la maschera prodotta dall'algoritmo.

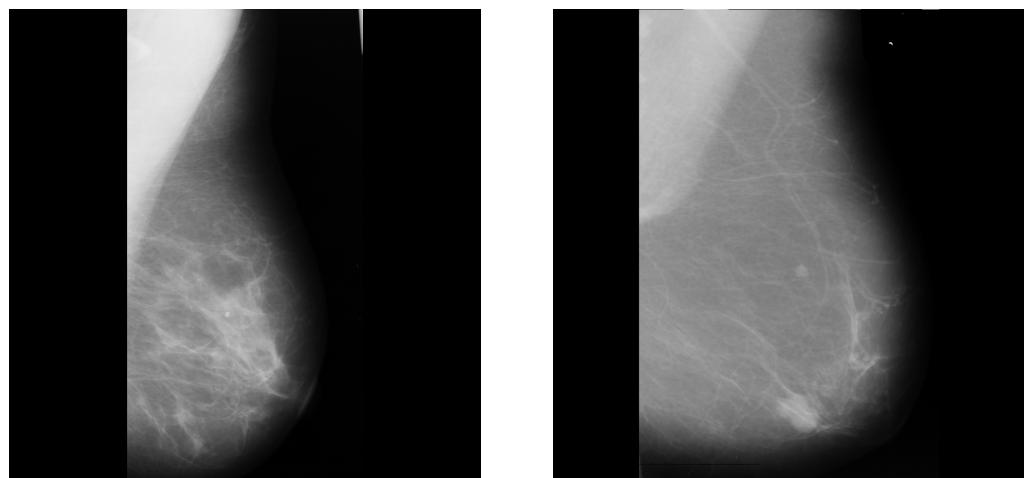


Figura 8: Immagini in input che presentano rispettivamente una piccola striscia bianche verticale e un piccolo cerchio bianco



Figura 9: Maschere prodotte dall'algoritmo, relative alla zona in cui sono presenti le regioni

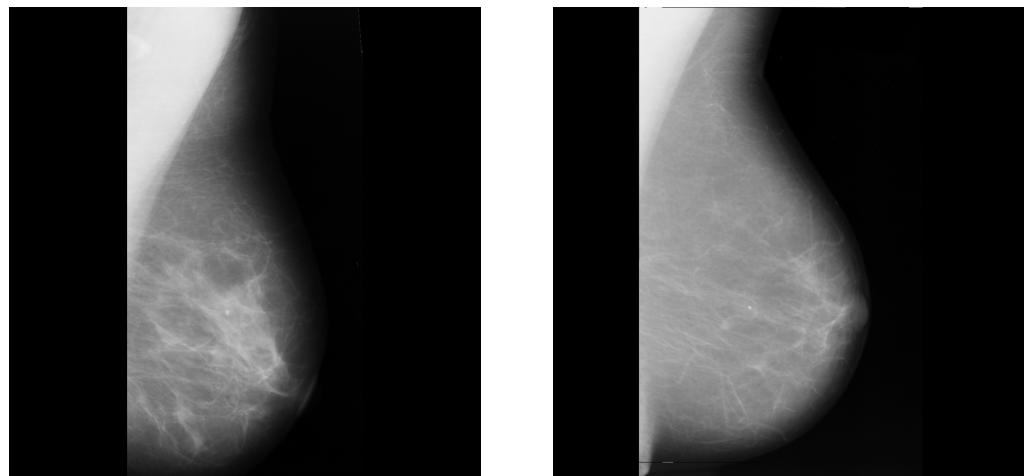


Figura 10: Output finale. Le strisce sono quasi completamente rimosse.

Casi particolari

Tre immagini rappresentano dei casi particolari, in cui si deve operare lungo entrambi gli assi e per cui gli algoritmi descritti non producono buoni risultati, pertanto vengono trattate al di fuori della pipeline descritta. Le tre immagini sono illustrate di seguito.

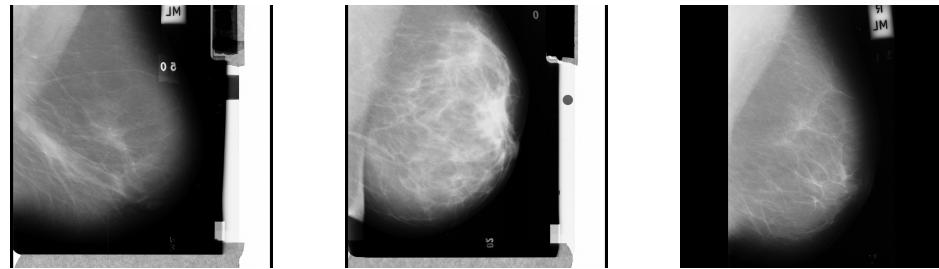


Figura 11: Immagini intrattabili dalla pipeline

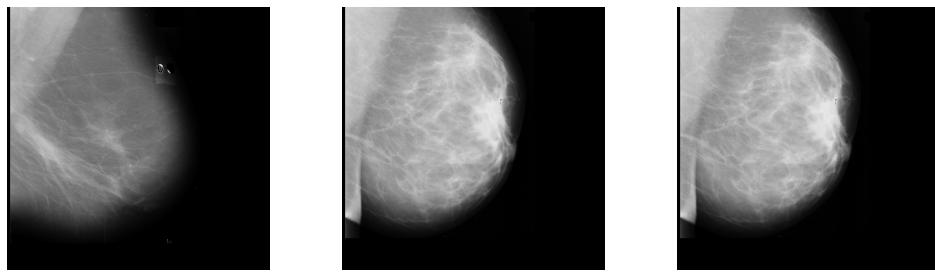


Figura 12: Immagini corrette

Quest'ultima operazione termina la pipeline, i cui risultati finali vengono descritti nella sezione successiva.

Risultati

La pipeline descritta è stata implementata all'interno del notebook jupyter main.ipynb, seguendo le sezioni descritte nella utilizzando le librerie Numpy, Pandas e OpenCV. L'esecuzione dell'intera pipeline richiede 26 secondi su una configurazione dotata di 32 GB di ram, CPU i7-10700 e sistema operativo Windows 11, utilizzando l'ambiente Visual Studio Code e IPython per eseguire le celle del notebook. Per visualizzare i risultati degli algoritmi, le immagini vengono salvate nel formato PNG. Con la pipeline descritta, le 211 immagini risultano prive, in alto a destra, dei blocchi di testo con sfondo bianco e di altri testi o singole lettere sparse. Sono state quasi del tutto eliminate piccole e grandi regioni di pixel bianchi. Pertanto, con l'applicazione degli algoritmi descritti, le immagini risultano più conformi e pronte per altri tipi di analisi. Di seguito viene mostrata l'applicazione della pipeline a ad alcune delle immagini del dataset. Come mostrato di seguito, la pipeline non rimuove del tutto alcuni artefatti.

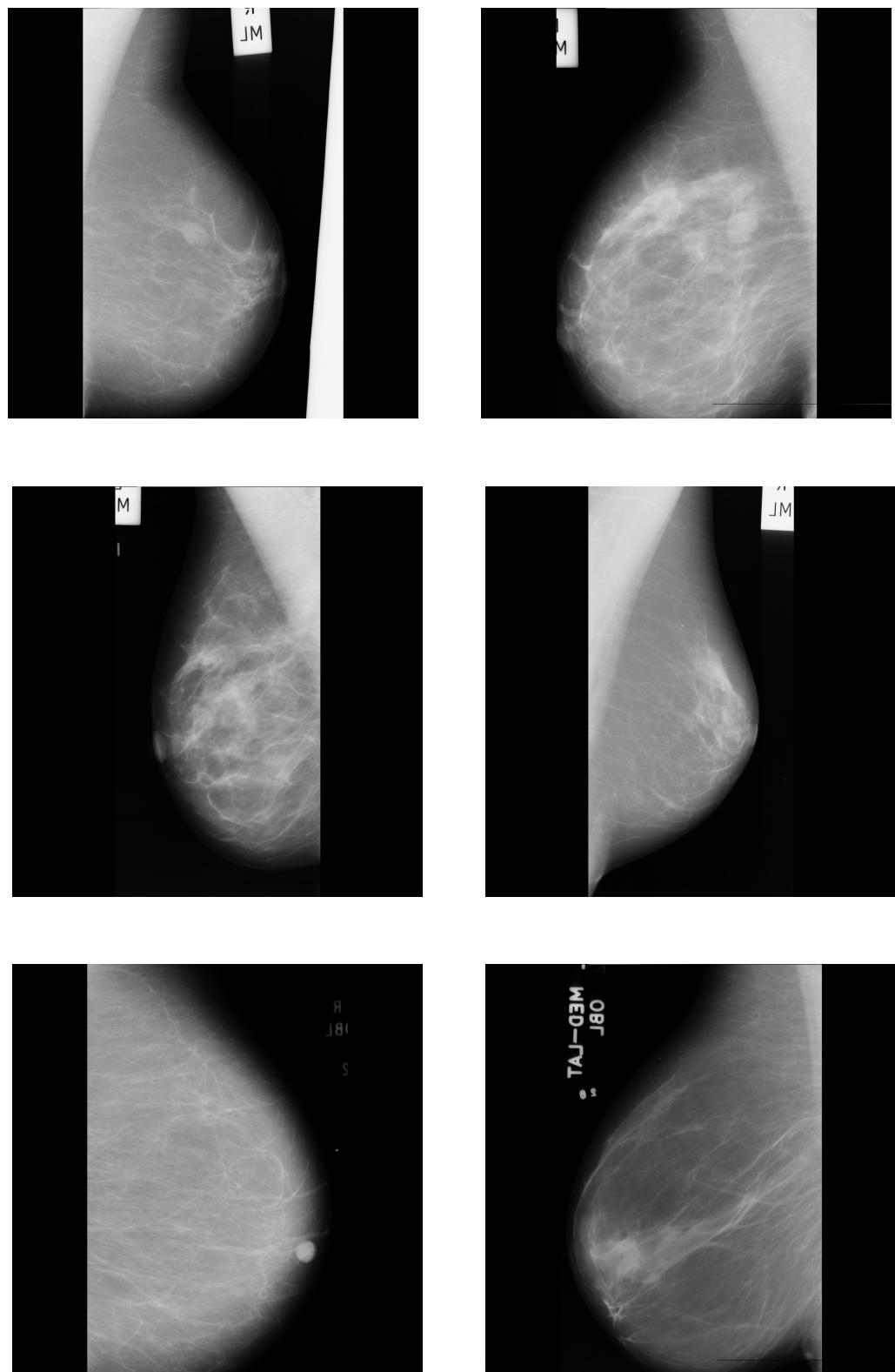


Figura 13: Immagini di input

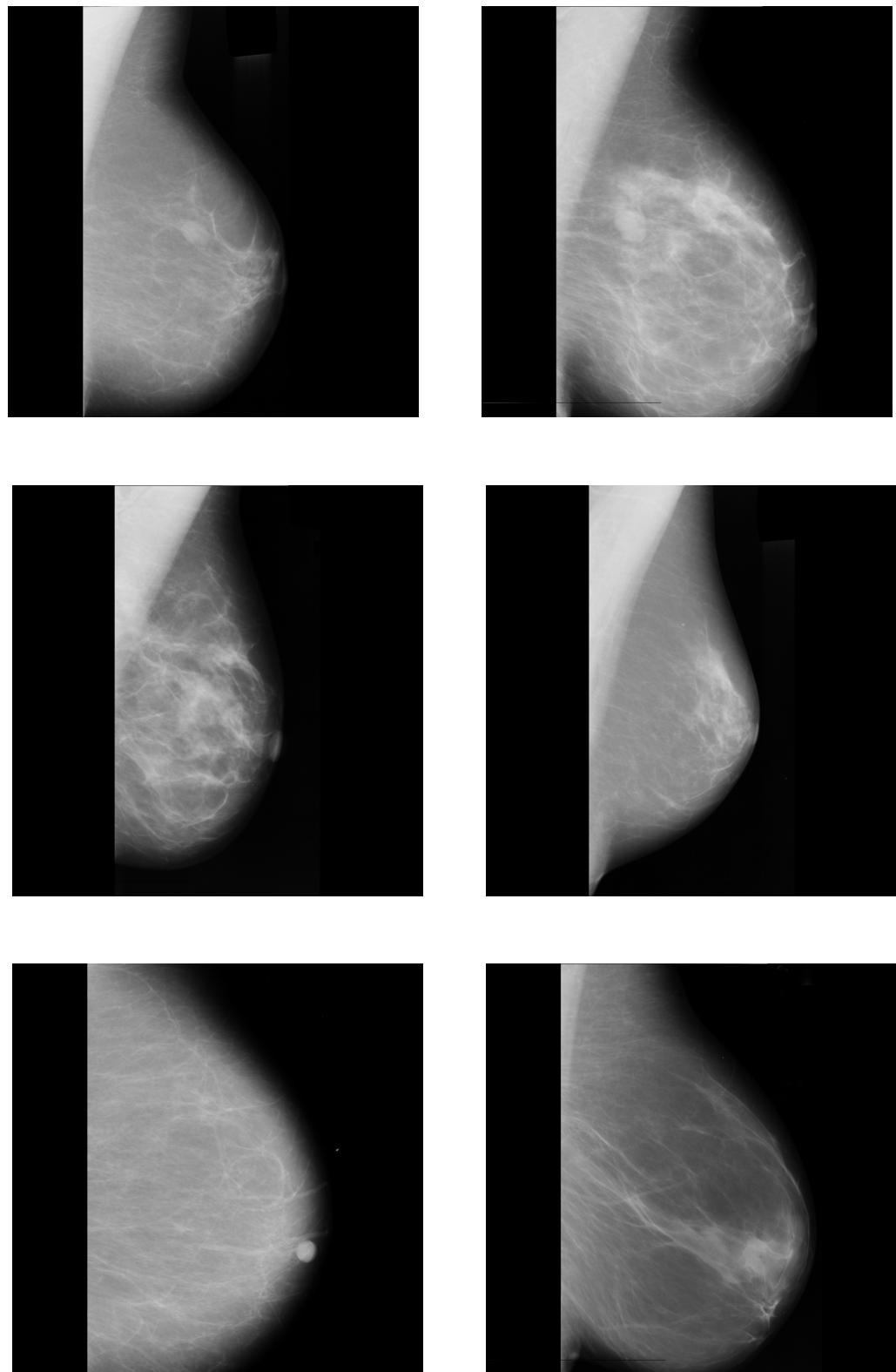


Figura 14: Immagini di output

Conclusione

Gli algoritmi descritti non risolvono tutte le problematiche presenti nelle mammografie. Vi sono alcuni casi particolari di artefatti, tra i quali: piccole lettere presenti nella cornice in basso a destra, linee orizzontali o verticali al centro, illuminazione non uniforme e rumore impulsivo a destra del seno. Visto l'esiguo numero di questi casi particolari e, considerato inoltre che ognuno di essi richiede diverse strategie di image processing, la miglior strategia di rimozione potrebbe essere quella trattarle i suddetti casi particolari separatamente, in modo simile a quanto effettuato per la rimozione di grandi strisce bianche.