

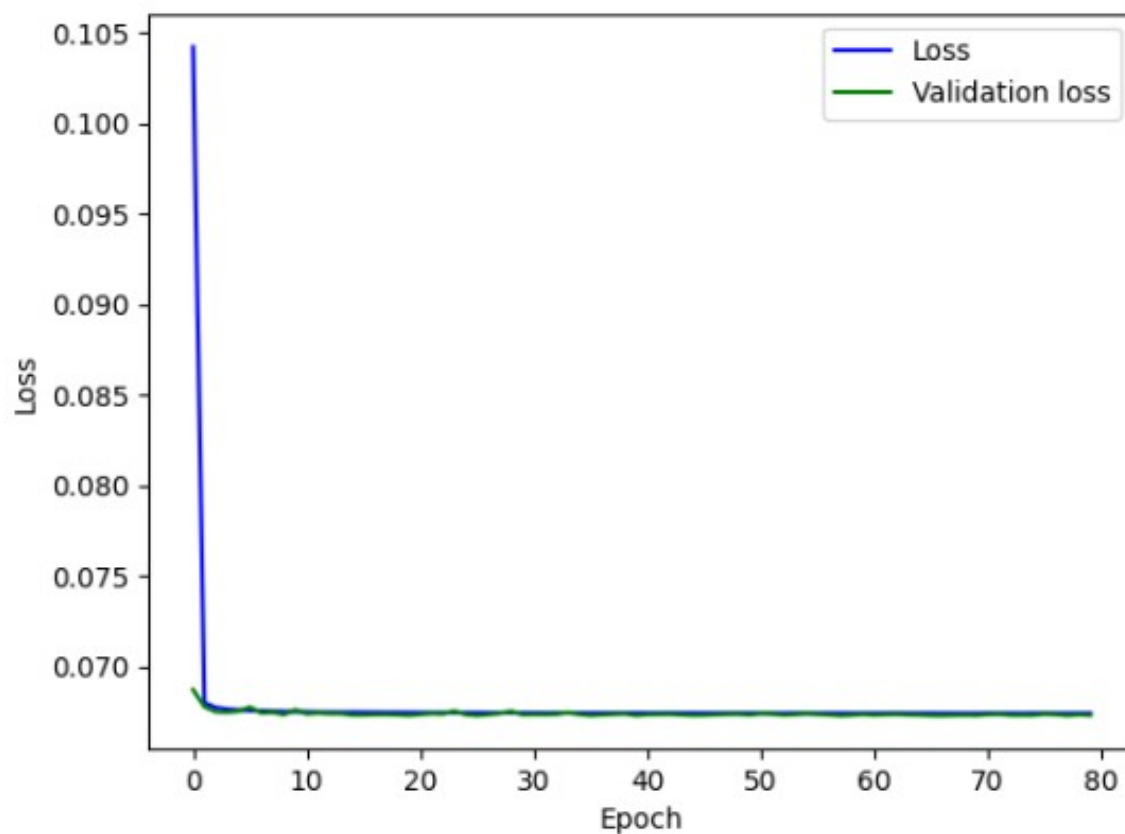
2. Παρατηρήσεις και Αποτελέσματα

A. reduce

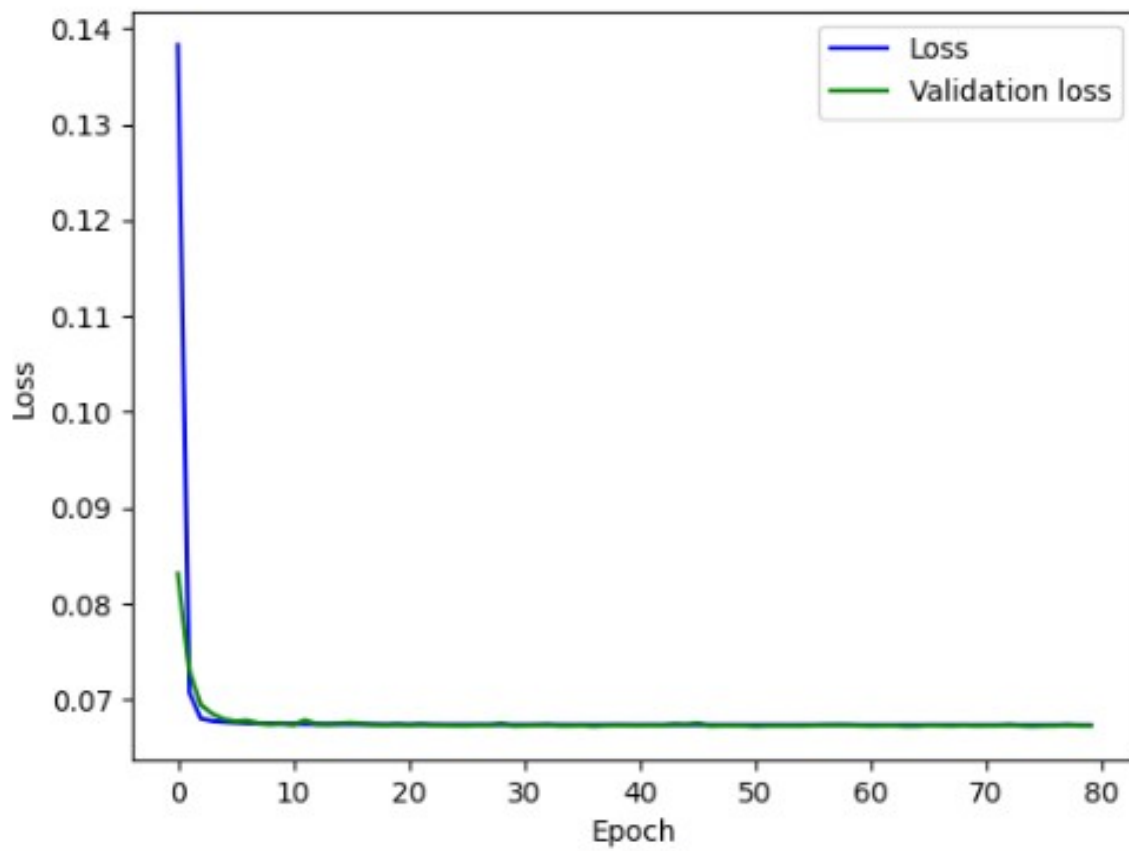
Έγιναν δοκιμές με διάφορες τιμές των υπερπαραμέτρων.

- Αριθμός layers:
Δοκιμάστηκαν όλοι οι αριθμοί έως το 7 (για μεγαλύτερους αριθμούς η εκτέλεση σταματούσε κάποια στιγμή γιατί τελείωνε τη μνήμη RAM). Παρατηρήσαμε τα βέλτιστα αποτελέσματα για τον αριθμό 6, καθώς για να τρέξουμε το πρόγραμμα με 7 έπρεπε να μειωθούν αισθητά οι υπόλοιπες υπερπαραμέτροι. Γενικά το loss μειωνόταν όσο αυξάνονταν τα layers.
- Μέγεθος filters:
Δοκιμάστηκαν διάφοροι αριθμοί. Παρατηρήσαμε τα βέλτιστα αποτελέσματα για τον αριθμό 32. Γενικά το loss μειωνόταν όσο αυξάνονταν το μέγεθος των filters μέχρι το 32, και μετά άρχισε να αυξάνεται το validation loss.
- Αριθμός filters:
Δοκιμάστηκαν διάφοροι μικροί αριθμοί. Παρατηρήσαμε τα βέλτιστα αποτελέσματα για τον αριθμό 7. Γενικώς μέχρι το 7 όσο αυξανόταν το πλήθος των filters τα αποτελέσματα βελτιωνόντουσαν, ενώ για αριθμούς >7, γινόταν overfitting.
- Αριθμός epochs:
Δοκιμάστηκαν διάφοροι αριθμοί έως το 100. Γενικώς η εναλλαγή του αριθμού των epochs δεν επηρέαζε σημαντικά τα αποτελέσματα. Όσο ο αριθμός αυτός αυξανόταν τα αποτελέσματα βελτιωνόντουσαν ελάχιστα. Τα βέλτιστα τα παρατηρήσαμε για 80. Μετά το 80 τα αποτελέσματα χειροτέρευαν ελάχιστα.
- Batch_size:
Δοκιμάστηκαν οι αριθμοί {64, 128, 256, 512}. Παρατηρήσαμε τα βέλτιστα αποτελέσματα για τον αριθμό 64. Γενικά όσο αυξάνεται το batch size παρατηρούμε μεγαλύτερη ταχύτητα στην εκτέλεση του προγράμματος αλλά παρουσιάζεται overfitting και αυξημένο loss. Με τους αριθμούς 64 και 128 φαίνεται να μην υπάρχει overfitting και έχουμε σταθερά χαμηλά αποτελέσματα σε loss/validation loss.
- Latency:
Δοκιμάστηκαν διάφοροι αριθμοί στο [1, 784] και παρατηρήθηκε το βέλτιστο αποτέλεσμα με διάσταση 5. Ξεκινήσαμε με την default τιμή (10). Κατεβαίνοντας μέχρι το 5 τα αποτελέσματα βελτιωνόντουσαν ενώ τα πιο χαμηλά νούμερα έβγαζαν χειρότερα αποτελέσματα. Για τιμές μεγαλύτερες του 10 τα αποτελέσματα ποικίλουν. Δεν υπήρχε κάποια σταθερή βελτίωση ή μείωση, οπότε δεν μπορούσαμε να βγάλουμε κάποιο γενικευμένο συμπέρασμα.

Στη συνέχεια περιλαμβάνονται ενδεικτικά διαγράμματα απο τα πειράματα που εκτελέσαμε. Τα πρώτα διαγράμματα παρουσιάζουν τα καλύτερα αποτελέσματα που επιτεύχθηκαν.

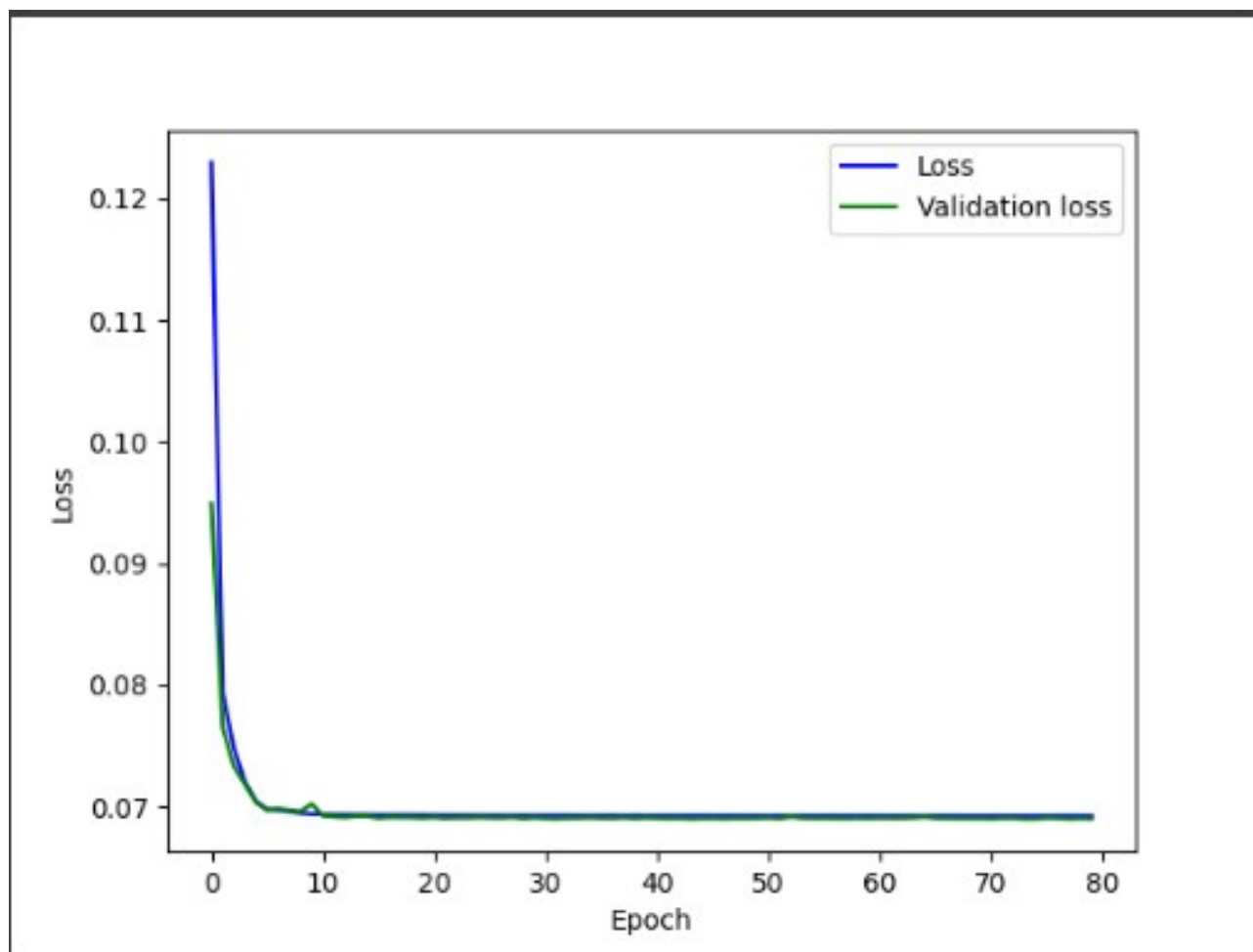


```
layer_num = 6  
filter_size = 32  
filter_num = 7  
epochs = 80  
batch_size = 64  
latent_space = 5
```

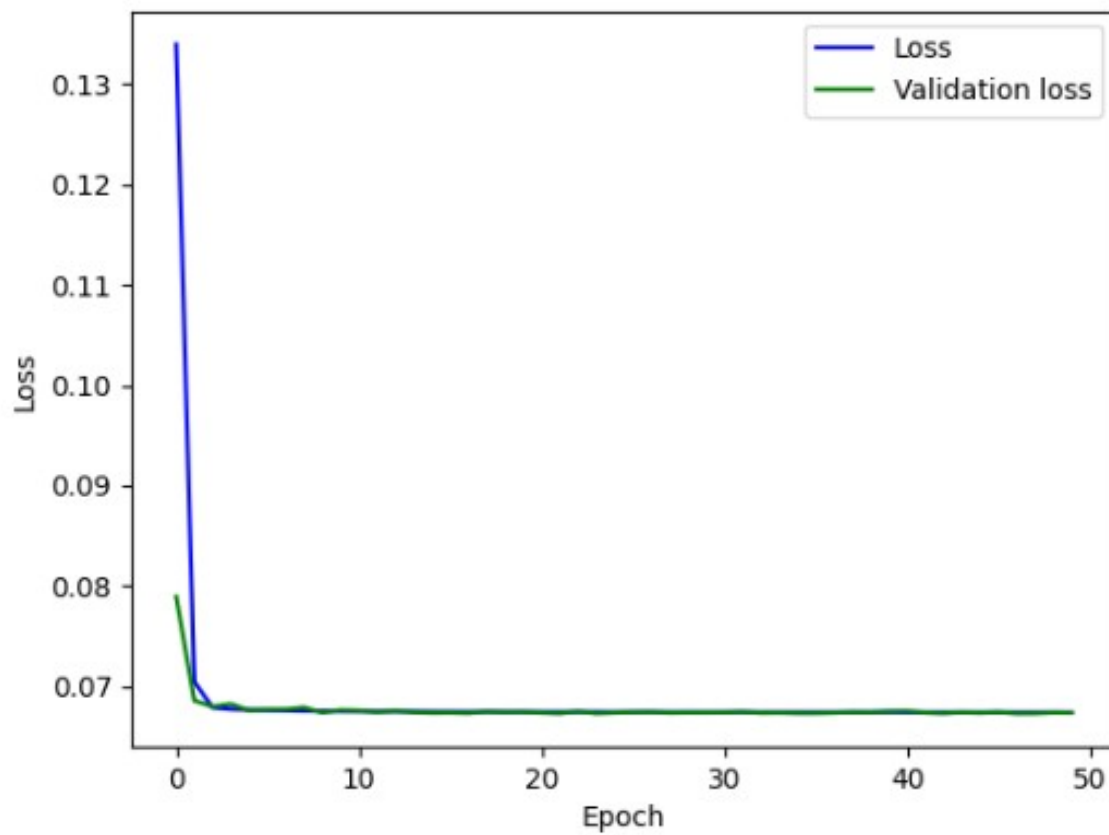


```
layer_num = 6  
filter_size = 32  
filter_num = 7  
epochs = 80  
batch_size = 128  
latent_space = 10
```

Έδω παρατηρούμε το αποτέλεσμα των παραμέτρων που προαναφέρθηκαν. Απο τα πρώτα 5 epoch βλέπουμε σταθερά βέλτιστα αποτελέσματα.

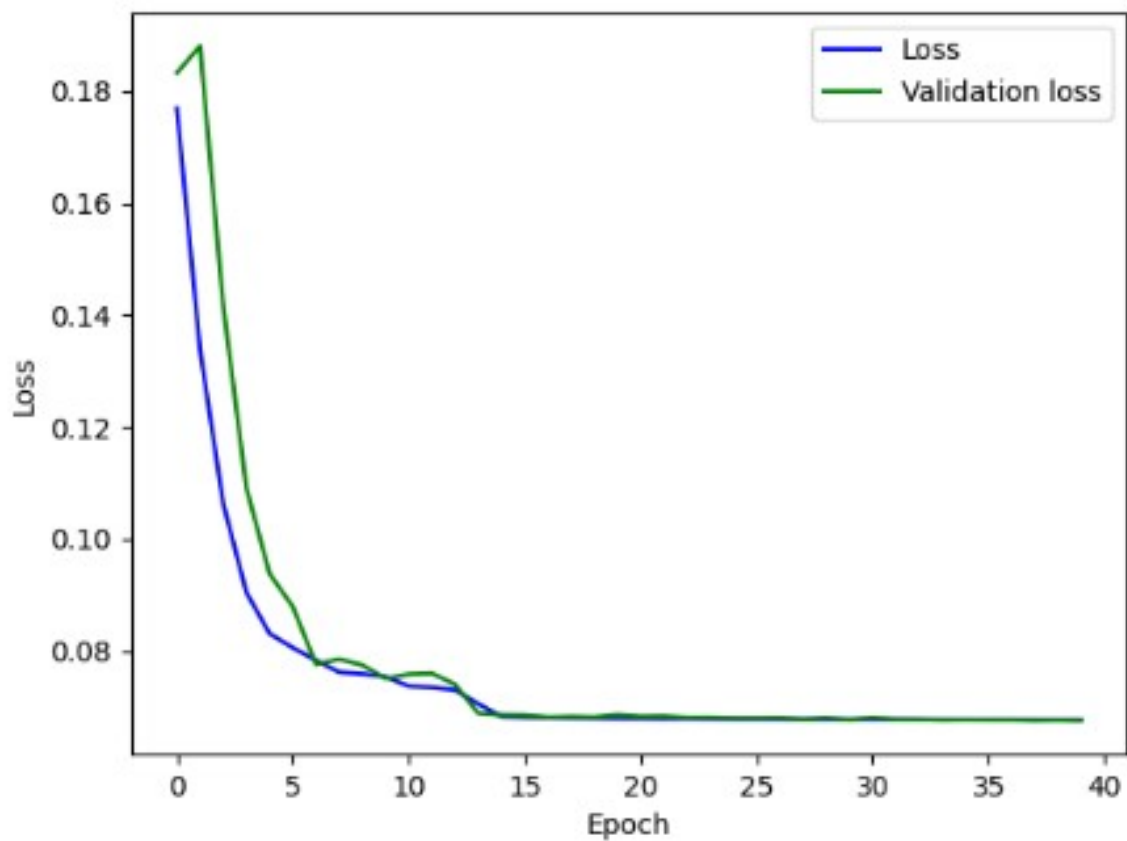


Η φωτογραφία αυτή δείχνει το αποτέλεσμα του προγράμματος με τις ίδιες υπερπαραμέτρους, είναι παρόμοιο με το βέλτιστο, αλλά εδώ βλέπουμε ότι στην αρχή το validation loss ήταν λίγο μεγαλύτερο.



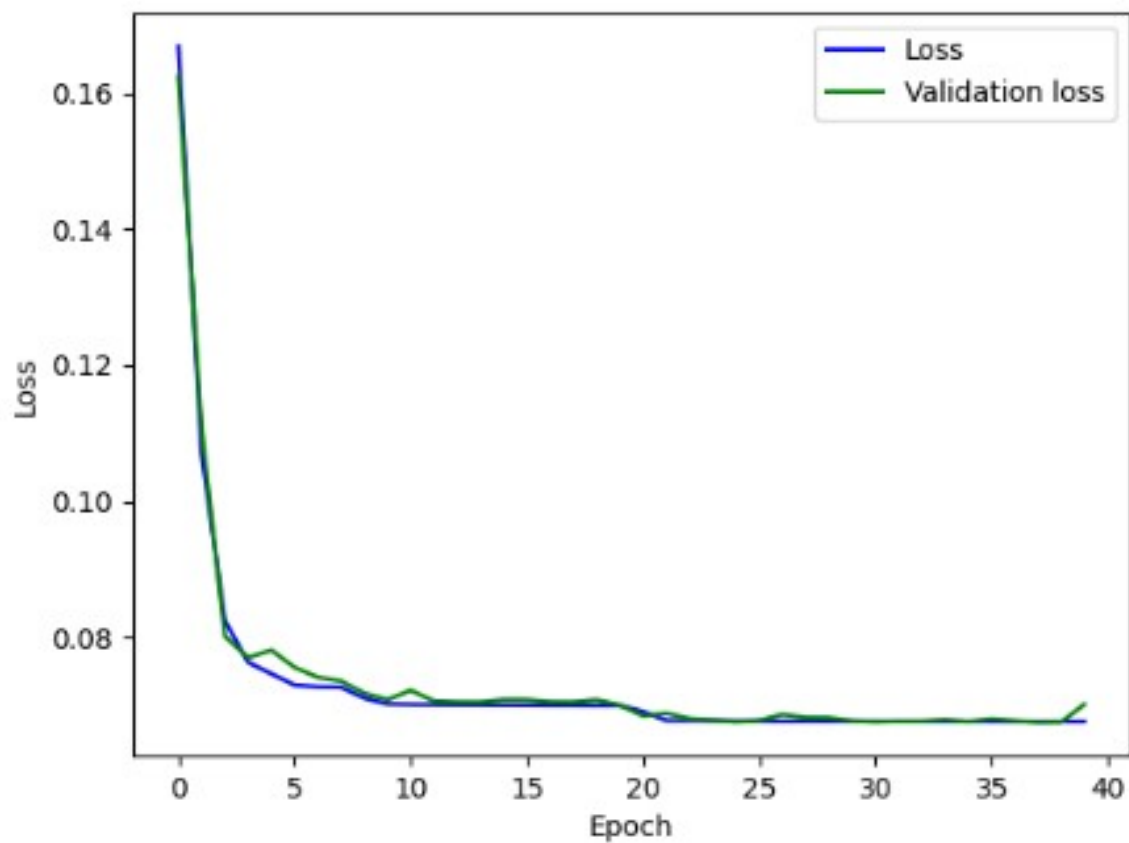
Έδω παραθέτουμε ένα ακόμα βέλτιστο αποτέλεσμα με το χαμηλότερο validation loss που καταφέραμε να επιτύχουμε.

```
layer_num = 6  
filter_size = 32  
filter_num = 7  
epochs = 50  
batch_size = 128  
latent_space = 5
```



Σε αυτή τη φωτογραφία βλέπουμε την επιρροή του μεγαλύτερου batch size στο πρόγραμμα και το πόσο αυξάνεται το validation loss, αν και φαίνεται να σταθεροποιείται κατά τη διάρκεια του προγράμματος. Οι παράμετροι για αυτό το πείραμα ήταν:

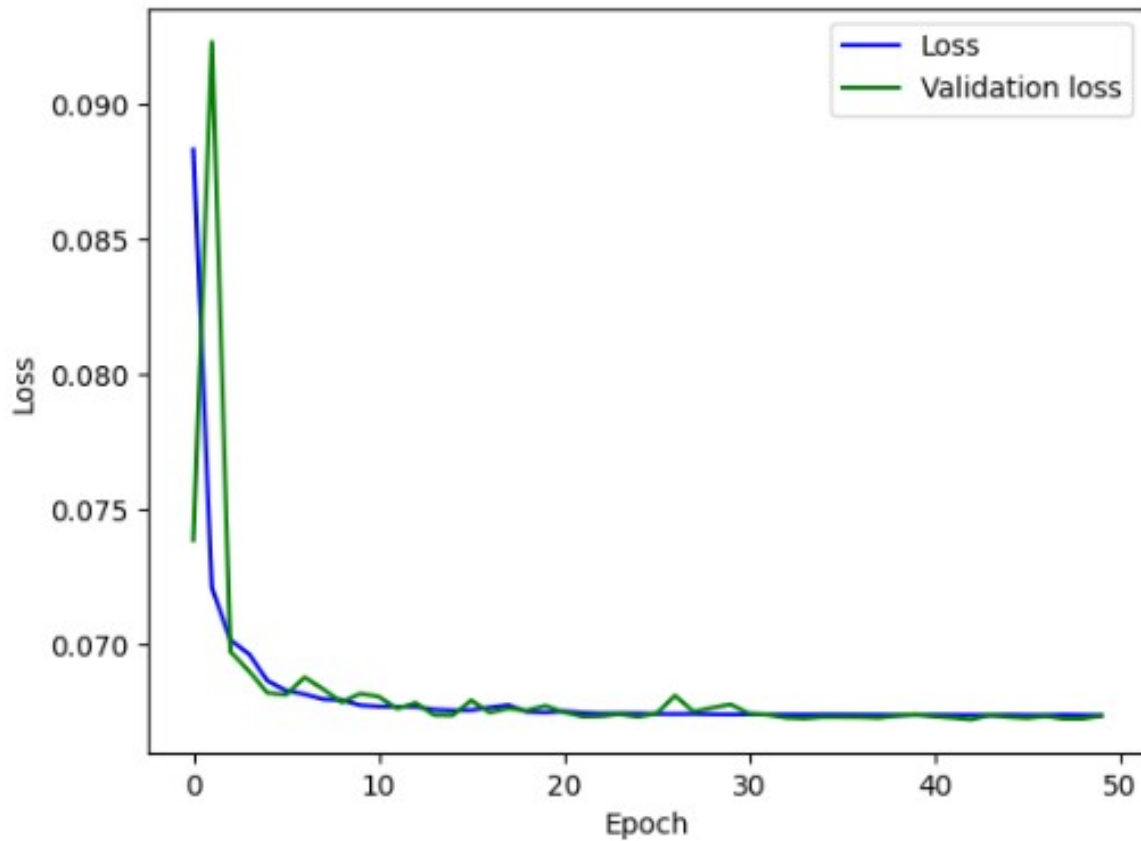
```
layer_num = 6  
filter_size = 32  
filter_num = 5  
epochs = 40  
batch_size = 512  
latent_space = 10
```



Ένα τελευταίο παράδειγμα με το batch size (=256) και τα αρνητικά αποτελέσματα που δημιουργεί.

```
layer_num = 6  
filter_size = 32  
filter_num = 5  
epochs = 40  
batch_size = 256  
latent_space = 10
```

Στη συνέχεια θα παρουσιάσουμε λίγα διαγράμματα με την επιρροή του filter size.

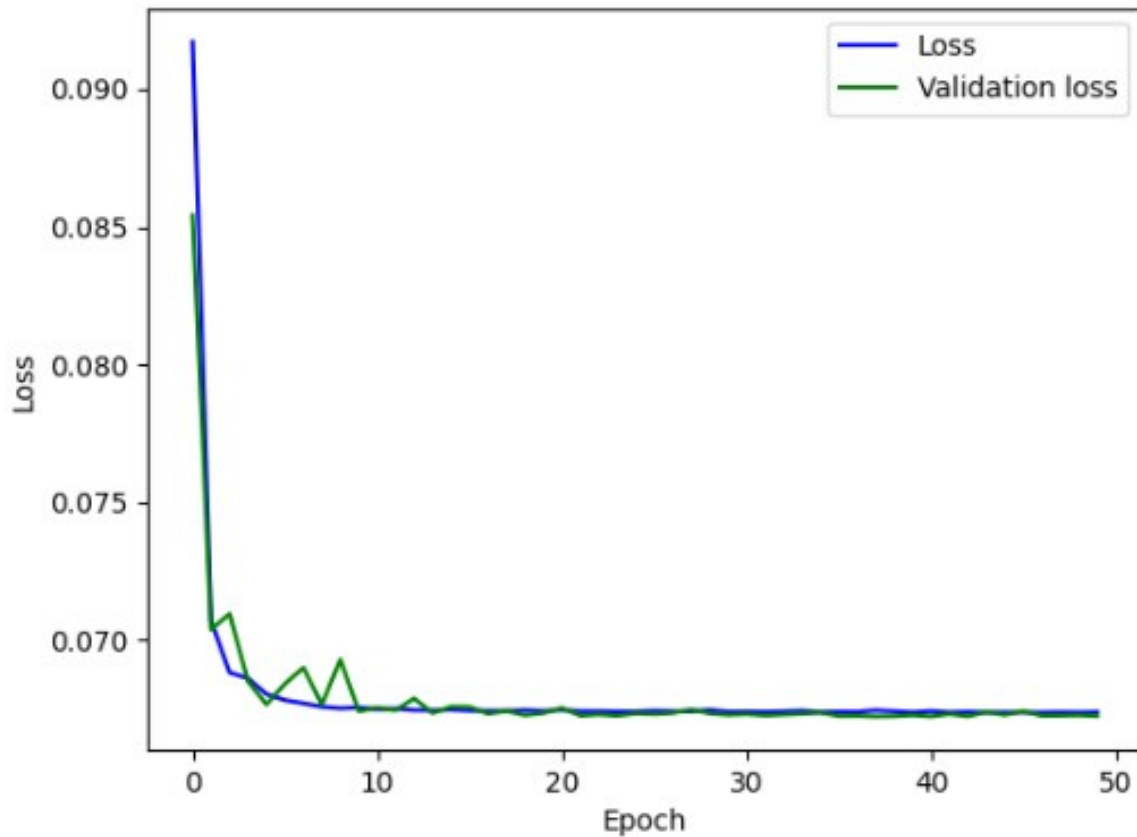


Οι παράμετροι που χρησιμοποιήθηκαν ήταν:

```
layer_num = 5  
filter_size = 128  
filter_num = 5  
epochs = 50  
batch_size = 128  
latent_space = 10
```

Μειώθηκαν τα layers λόγω έλλειψης μνήμης ώστε να μπορέσουμε να δοκιμάσουμε μεγαλύτερο filter size. Παρατηρούμε ότι τα αποτελέσματα αλλοιώθηκαν με την μεγάλη αύξηση του filter_size.

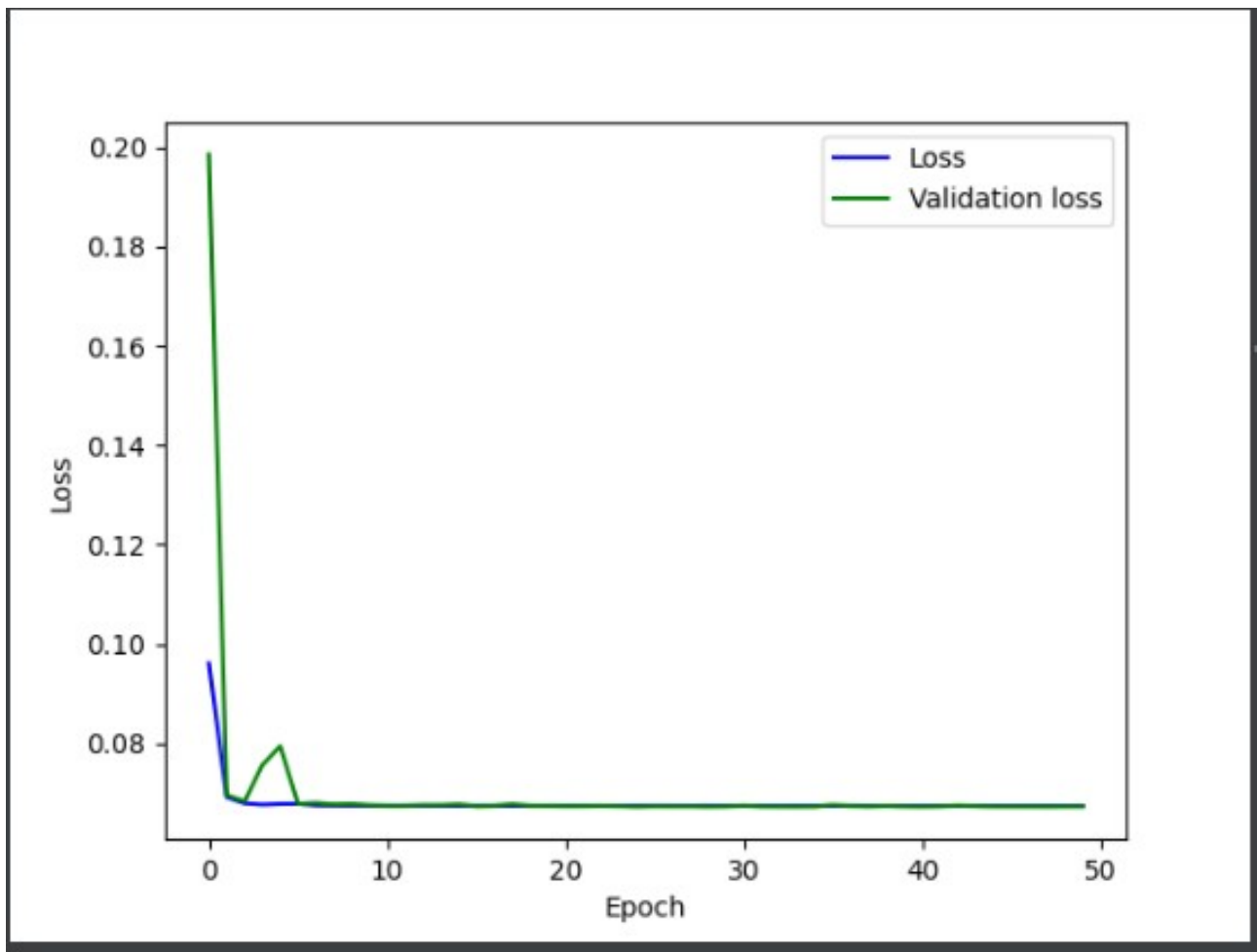
Στη συνέχεια θα προσθεθούν δυο παραδείγματα με αυξημένο filter size (=64) σε συνδυασμό αύξησης του αριθμού των φίλτρων.



Παρατηρούμε ότι το τελικό validation loss φτάνει σε πολύ χαμηλές τιμές, όμως στα πρώτα epochs υπάρχει overfitting.

Οι παράμετροι που χρησιμοποιήθηκαν ήταν:

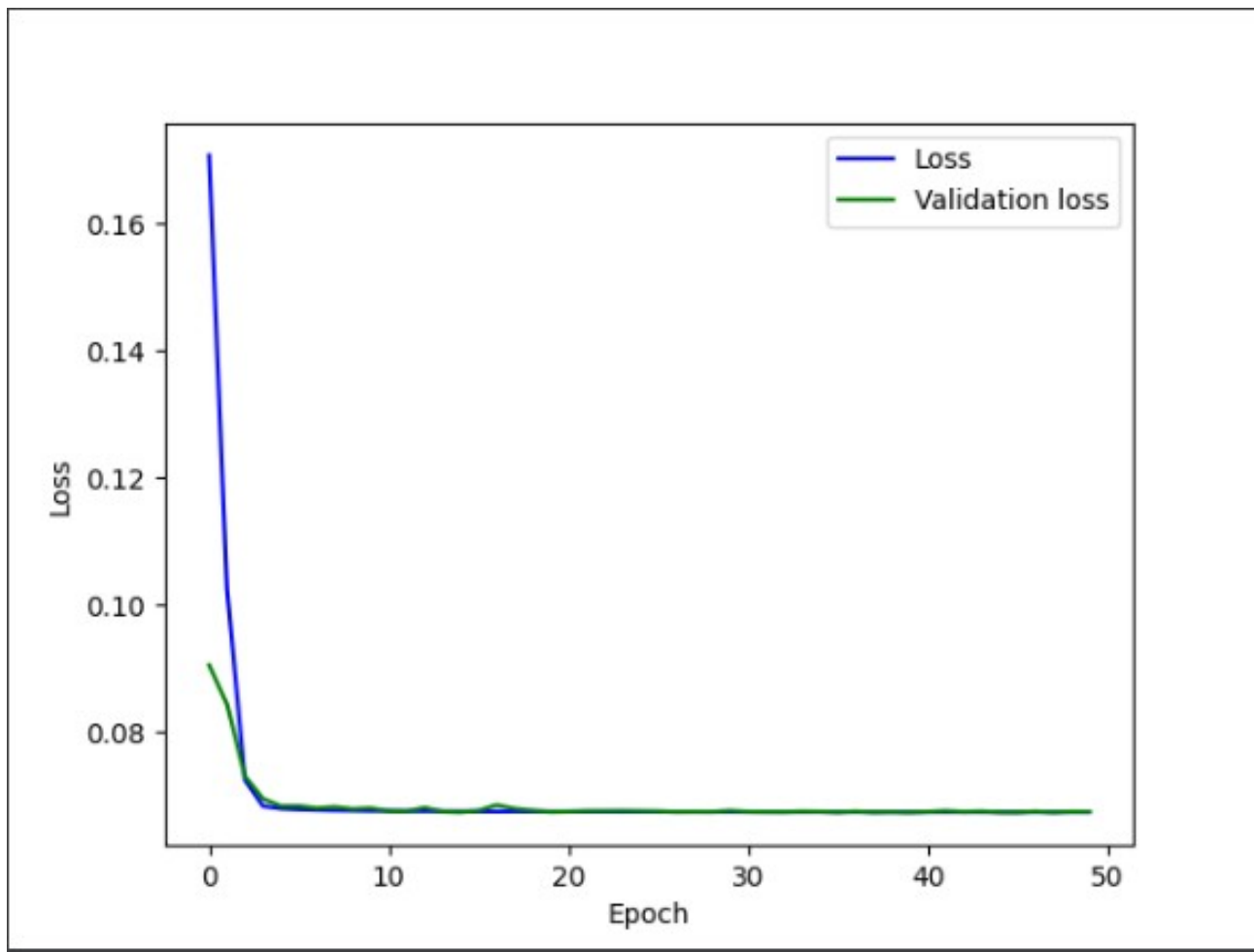
```
layer_num = 5  
filter_size = 128  
filter_num = 9  
epochs = 50  
batch_size = 128  
latent_space = 10
```



Μειώνοντας το `filter_size` απο το προηγούμενο πείραμα παρατηρούμε ότι το overfitting μειώθηκε αλλά δεν εξαφανίσθηκε.

Οι παράμετροι που χρησιμοποιήθηκαν ήταν:

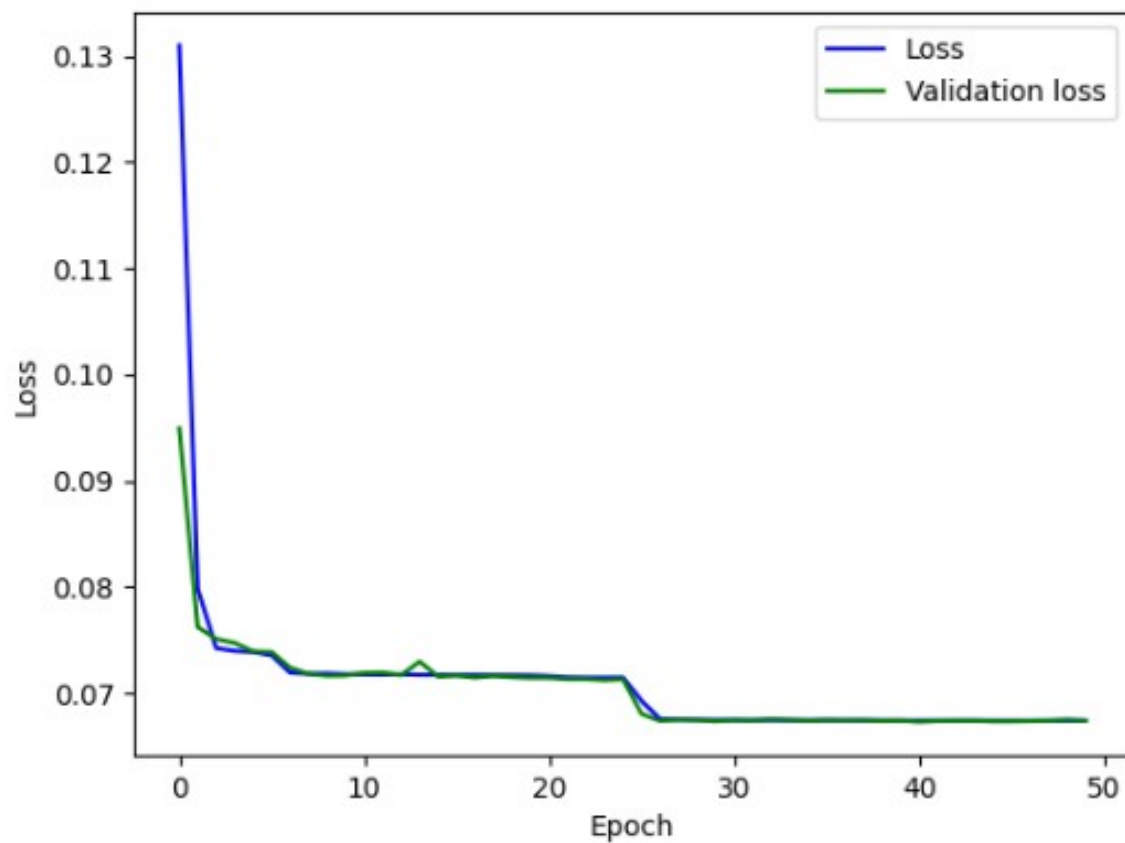
```
layer_num = 5  
filter_size = 64  
filter_num = 9  
epochs = 50  
batch_size = 128  
latent_space = 10
```



Παρουσιάζουμε ένα ακόμα απο τα παραδείγματα με διάσταση 5.

Οι παράμετροι που χρησιμοποιήθηκαν ήταν:

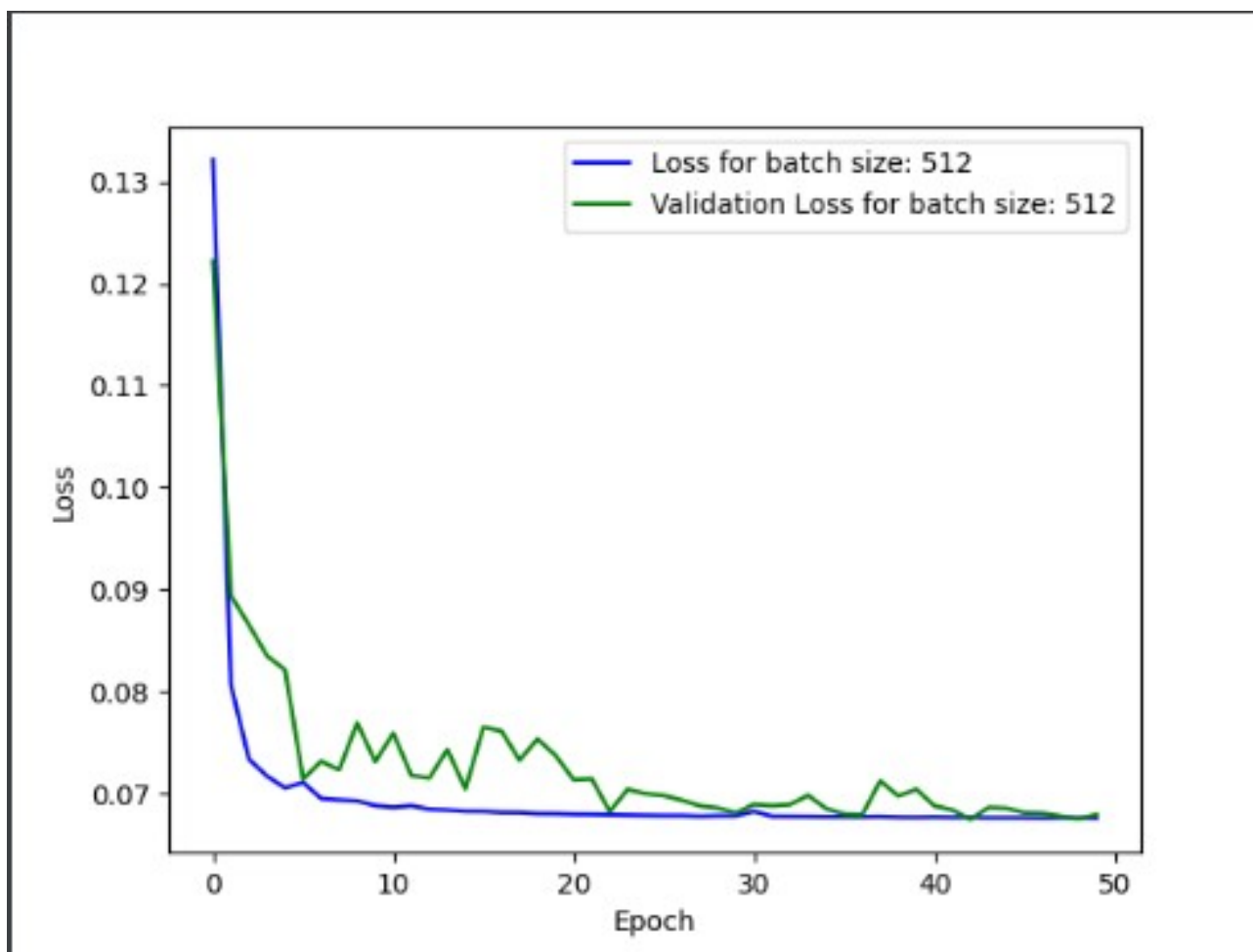
```
layer_num = 6  
filter_size = 32  
filter_num = 7  
epochs = 50  
batch_size = 256  
latent_space = 5
```



Παραθέτουμε ένα παράδειγμα με τον πειραματισμό στον αριθμό των διαστάσεων με όχι τόσο καλά αποτελέσματα.

Οι παράμετροι που χρησιμοποιήθηκαν ήταν:

```
layer_num = 6  
filter_size = 32  
filter_num = 7  
epochs = 50  
batch_size = 128  
latent_space = 100
```



Τέλος θα παρουσιάσουμε ένα από τα αρχικά αποτελέσματα του προγράμματος πριν αντιμετωπιστεί το πρόβλημα του Overfitting, το οποίο διορθώθηκε με την χρήση των Dropout layers.

B. search

Ο χρόνος σε όλα τα πειράματα ήταν σημαντικά καλύτερος με τον reduced LSH από ότι με τον true.

Το approximation factor, ήταν αρκετά μικρότερο στον reduced από ότι στον true για μικρές διαστάσεις, ενώ μετά από κάποια τιμή έγινε μεγαλύτερο.

Αποτελέσματα για latent_dim = 5:

tReduced: 0.000865
tLSH: 0.034855
tTrue: 0.162266
Approximation Factor LSH: 73.713203
Approximation Factor Reduced: 7.895600

Αποτελέσματα για latent_dim = 10:

tReduced: 0.001076
tLSH: 0.028861
tTrue: 0.141261
Approximation Factor LSH: 145.949997
Approximation Factor Reduced: 17.389999

Αποτελέσματα για latent_dim = 50:

tReduced: 0.004394
tLSH: 0.033670
tTrue: 0.158045
Approximation Factor LSH: 73.871002
Approximation Factor Reduced: 94.042801