

SVD και Επεξεργασία Εικόνας (Image Compression)

14 Μαΐου 2025

Η Ανάλυση Ιδιάζουσων Τιμών (Singular Value Decomposition - SVD) είναι μια ισχυρή τεχνική γραμμικής άλγεβρας που χρησιμοποιείται ευρέως στην επεξεργασία εικόνας, ειδικά για τη συμπίεση εικόνας.

Η SVD είναι μια μέθοδος που διασπά έναν πίνακα σε τρεις πίνακες:

$$A = U\Sigma V^T$$

έναν ορθογώνιο πίνακα U , έναν διαγώνιο πίνακα Σ και τον ανάστροφο ορθογώνιο πίνακα V^T .

Υπενθυμίζουμε ότι αν A είναι ένας πραγματικός $m \times n$ πίνακας, τότε υπάρχουν ορθογώνιοι πίνακες U και V έτσι ώστε

$$A_{m \times n} = U_{m \times m} \begin{bmatrix} \Sigma_{m \times n} & 0 \\ 0 & 0 \end{bmatrix} V_{n \times n}^T = U_{m \times m} \Sigma_{m \times n} V_{n \times n}^T$$

όπου Σ ένας μη ιδιάζων διαγώνιος πίνακας, με θετικά διαγώνια στοιχεία διατεταγμένα κατά φθίνουσα σειρά. Ο αριθμός των μη μηδενικών διαγώνιων στοιχείων του Σ ισούται με την τάξη του πίνακα A .

Η ανάλυση $A = U\Sigma V^T$ είναι γνωστή σαν ανάλυση ιδιάζουσων τιμών (Singular Value Decomposition - SVD) του πίνακα A .

1 Εφαρμογή στην Επεξεργασία Εικόνας

Μια εικόνα μπορεί να αναπαρασταθεί ως πίνακας όπου κάθε στοιχείο αντιστοιχεί σε ένα pixel. Εφαρμόζοντας την SVD στον πίνακα της εικόνας, λαμβάνουμε τους πίνακες $A = U\Sigma V^T$.

Στόχος: Μείωση των Διαστάσεων

Για τη συμπίεση, μπορούμε να διατηρήσουμε μόνο τις μεγαλύτερες ιδιάζουσες τιμές στον πίνακα Σ . Αυτή η τακτική μειώνει την ποσότητα των δεδομένων που χρειάζονται για την αναπαράσταση της εικόνας. Στο πλαίσιο της ανακατασκευής της εικόνας, μπορούμε να προσεγγίσουμε τον αρχικό πίνακα A με έναν πίνακα $A_k = U\Sigma_k V^T$.

Ανακατασκευή της Εικόνας:

Ο A_k , χρησιμοποιώντας τη δυνατότητα ανάκτησης των κομματιών της SVD με πίνακες τάξης

1, μπορεί να γραφτεί:

$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^T = \sigma_1 u_1 v_1^T + \dots + \sigma_k u_k v_k^T$$

Ειδικότερα, αν υποθέσουμε ότι έχουμε μια εικόνα ανάλυσης $640 \times 597 = 382,080$ **pixels** και προσπαθήσουμε να προσεγγίσουμε τον πίνακα της εικόνας με προσέγγιση $k = 50$, θα καταλήξουμε σε ανάλυση εικόνας $(640 + 597) \times 50 = 61,850$ **pixels**, επιτυγχάνοντας ποσοστό συμπίεσης **16,19%** ($= 61,850/382,080$), λαμβάνοντας πάντα υπόψη τη διατήρηση ποιότητας. Η ποιότητα της εικόνας μπορεί να ελεγχθεί διατηρώντας περισσότερες ή λιγότερες ιδιάζουσες τιμές.

2 Απαιτούμενη Μνήμη και Ποσοστό Συμπίεσης

Για να αξιολογήσουμε την αποτελεσματικότητα της συμπίεσης εικόνας μέσω της Ανάλυσης Ιδιαζουσών Τιμών (SVD), είναι σημαντικό να υπολογίσουμε την απαιτούμενη μνήμη πριν και μετά τη συμπίεση, καθώς και το ποσοστό συμπίεσης. Αυτοί οι υπολογισμοί εξαρτώνται από τις διαστάσεις της εικόνας $m \times n$ και τον βαθμό προσέγγισης k .

2.1 Απαιτούμενη Μνήμη

Ασπρόμαυρη Εικόνα

Μια αρχική εικόνα, είτε ασπρόμαυρη είτε έγχρωμη, αναπαριστάται ως ένας πίνακας με διαστάσεις $m \times n$. Για μια **ασπρόμαυρη εικόνα**, κάθε pixel αποθηκεύεται ως ένας αριθμός (π.χ., float64), συνεπώς η συνολική μνήμη που απαιτείται είναι:

$$m \times n$$

Έγχρωμη Εικόνα

Για μια **έγχρωμη εικόνα**, η μνήμη διαιρείται σε τρία κανάλια χρώματος RGB¹, επομένως:

$$3 \times m \times n$$

2.2 Απαιτούμενη Μνήμη μετά τη Συμπίεση με SVD

Η συμπίεση μέσω SVD περιλαμβάνει την αποθήκευση των τριών πινάκων U , Σ και V^T , όπου:

- Ο πίνακας U έχει διαστάσεις $m \times k$
- Ο διαγώνιος πίνακας Σ μπορεί να αναπαρασταθεί ως διανύσμα με k στοιχεία

¹RGB (Red, Green, Blue) είναι ένα μοντέλο που χρησιμοποιείται για την αναπαράσταση εικόνων με βάση τα χρώματα: Κόκκινο, Πράσινο και Μπλε. Κάθε pixel σε μια εικόνα RGB αναπαρίσταται από τρεις τιμές, που αντιστοιχούν στις εντάσεις του κόκκινου, του πράσινου και του μπλε. Συνδυάζοντας αυτά τα τρία χρώματα με διάφορους τρόπους, μπορούμε να παράγουμε ένα ευρύ φάσμα χρωμάτων. Για μια έγχρωμη εικόνα, η SVD μπορεί να εφαρμοστεί ξεχωριστά σε κάθε κανάλι χρώματος (R, G, B), δηλαδή εφαρμόζεται ξεχωριστά σε κάθε έναν από αυτούς τους πίνακες. Αυτό έχει ως αποτέλεσμα τρία σύνολα πινάκων U , Σ και V^T .

- Ο πίνακας V^T έχει διαστάσεις $k \times n$

Συνεπώς, η συνολική μήμη που απαιτείται μετά τη συμπίεση είναι:

Ασπρόμαυρη Εικόνα

$$\text{Μνήμη}_{\text{συμπίεσμένη}} = (m \times k) + k + (n \times k) = k(m + n + 1)$$

Έγχρωμη Εικόνα

Για μια έγχρωμη εικόνα, η διαδικασία αυτή εφαρμόζεται σε κάθε ένα από τα τρία κανάλια χρώματος, άρα:

$$\text{Μνήμη}_{\text{συμπίεσμένη}} = 3 \times k(m + n + 1)$$

2.3 Ποσοστό Συμπίεσης

Το ποσοστό συμπίεσης (Compression Rate) ορίζεται ως ο λόγος της μνήμης μετά τη συμπίεση προς τη μνήμη πριν τη συμπίεση, πολλαπλασιασμένος επί 100 .

Ασπρόμαυρη Εικόνα

$$\text{Ποσοστό Συμπίεσης} = \left(\frac{k(m + n + 1)}{m \times n} \right) \times 100\%$$

Έγχρωμη Εικόνα

$$\text{Ποσοστό Συμπίεσης} = \left(\frac{3 \times k(m + n + 1)}{3 \times m \times n} \right) \times 100\% = \left(\frac{k(m + n + 1)}{m \times n} \right) \times 100\%$$

2.4 Παράδειγμα Υπολογισμού

Ασπρόμαυρη Εικόνα

Ας υποθέσουμε ότι έχουμε μια ασπρόμαυρη εικόνα ανάλυσης $640 \times 597 = 382,080$ pixels και χρησιμοποιούμε προσέγγιση με $k = 50$.

- **Αρχική Μνήμη:** $640 \times 597 = 382,080$ pixels
- **Συμπίεσμένη Μνήμη:** $50 \times (640 + 597 + 1) = 50 \times 1238 = 61,900$ pixels
- **Ποσοστό Συμπίεσης:** $\left(\frac{61,900}{382,080} \right) \times 100\% \approx 16.19\%$

Αυτό σημαίνει ότι η συμπίεσμένη εικόνα καταλαμβάνει περίπου το 16.19% της αρχικής μνήμης, επιτυγχάνοντας σημαντική μείωση του μεγέθους χωρίς σημαντική απώλεια ποιότητας.

Έγχρωμη Εικόνα

Ας υποθέσουμε τώρα ότι έχουμε μια **έγχρωμη εικόνα** ανάλυσης $640 \times 597 = 382,080$ pixels . Σε μια έγχρωμη εικόνα, κάθε pixel αποτελείται από τρία κανάλια χρώματος (RGB), επομένως ο συνολικός αριθμός των δεδομένων πριν τη συμπίεση είναι:

- **Αρχική Μνήμη:** $3 \times 640 \times 597 = 1,146,240$ pixels

Εφαρμόζοντας την SVD με βαθμό προσέγγισης $k = 50$ σε κάθε ένα από τα τρία κανάλια χρώματος, η συνολική μνήμη που απαιτείται μετά τη συμπίεση υπολογίζεται ως:

$$\text{Μνήμη}_{\text{συμπιεσμένη}} = 3 \times k(m+n+1) = 3 \times 50 \times (640+597+1) = 3 \times 50 \times 1238 = 185,700 \text{ pixels}$$

Ποσοστό Συμπίεσης:

Το ποσοστό συμπίεσης υπολογίζεται ως ο λόγος της μνήμης μετά τη συμπίεση προς τη μνήμη πριν τη συμπίεση, πολλαπλασιασμένος επί 100 για να εκφραστεί σε %.

$$\text{Ποσοστό Συμπίεσης} = \left(\frac{185,700}{1,146,240} \right) \times 100\% \approx 16.19\%$$

- **Αρχική Μνήμη:** 1,146,240 pixels
- **Συμπιεσμένη Μνήμη:** 185,700 pixels
- **Ποσοστό Συμπίεσης:** 16.19%

2.5 Συμπεράσματα

Όπως παρατηρήθηκε και στο παράδειγμα της ασπρόμαυρης εικόνας, η συμπίεση μέσω της SVD επιτυγχάνει το ίδιο ποσοστό συμπίεσης (16.19%) για έγχρωμες εικόνες. Αυτό συμβαίνει επειδή ο υπολογισμός του ποσοστού συμπίεσης είναι ανεξάρτητος από τον αριθμό των καναλιών χρώματος, καθώς ο παράγοντας 3 (για τα RGB κανάλια) απλοποιείται στο ποσοστό.

Αυτό επιβεβαιώνει ότι η μέθοδος της SVD είναι εξίσου αποτελεσματική τόσο για ασπρόμαυρες όσο και για έγχρωμες εικόνες, παρέχοντας σημαντική μείωση του μεγέθους των δεδομένων χωρίς σημαντική απώλεια ποιότητας.

3 Υλοποίηση με JULIA Image Compression using SVD

3.1 Ασπρόμαυρη Εικόνα

2

```
using Images, FileIO, LinearAlgebra, Plots

# Specify the path to your local image file
img_path = "C:\\Users\\yanis\\Downloads\\cats_medium.jpg"
img = load(img_path)
img_gray = float.(Gray.(img))

function rank_approx(F::SVD, k)
    U, S, V = F
    # Ensure k does not exceed the dimensions of U and V
    k = min(k, size(U, 2), size(V, 2))
    # Calculate the approximation of the original matrix using
    # the first k factors.
    # This is done by multiplying the first k columns of U,
    # the first k elements of the diagonal matrix S,
    # and the first k rows of V'
    M = U[:, 1:k] * Diagonal(S[1:k]) * V[:, 1:k]'
    # Ensure all values of the matrix M are between 0 and 1.
    # This is important for image display, as pixel values must be
    # in this range.
    clamp01!(M)
    return M
end
```

²Επεξήγηση ορισμένων σημείων του κώδικα αναφορικά με τη μετατροπή της εικόνας σε πίνακα και το αντίθετο:

Η συνάρτηση `load`, η οποία περιέχεται στο `FileIO` package, φορτώνει οποιοδήποτε αρχείο εικόνας μετατρέποντάς το σε πίνακα του οποίου κάθε στοιχείο είναι ένα χρώμα. Ειδικότερα, σε μία εικόνα διαστάσεων $m \times n$, ο `img` είναι ένας πίνακας διάστασης $m \times n$. Επισημαίνουμε ότι κάθε στοιχείο του πίνακα αντιστοιχεί στα τρία βασικά χρώματα (κόκκινο, πράσινο και μπλε) ή αλλιώς κανάλια χρωμάτων όπως αναφέρονται. Εν συνεχεία, μέσω της συνάρτησης `Gray`, κάθε πίξελ μετατρέπεται σε πιξελ αποχρώσεων του γκρι χρώματος βάσει της στάθμησης των τριών βασικών χρωμάτων που αποτελούν το αρχικό πιξελ. Έτσι, το αποτέλεσμα είναι ένας νέος πίνακας, κάθε στοιχείο του οποίου είναι απόχρωση του γκρι χρώματος. Κατόπιν, μέσω της συνάρτησης `float`, κάθε στοιχείο του πίνακα μετατρέπεται σε στοιχείο κινητής υποδιαστολής.

Ανάλυση SVD – k rank:

Η συνάρτηση `svd`, ενσωματωμένη συνάρτηση στο πακέτο `LinearAlgebra`, κατά τα γνωστά από τη θεωρία, παραγοντοποιεί τον πίνακα σε δύο ορθογώνιους πίνακες και σε έναν διαγώνιο με στοιχεία τις ιδιοτιμές του πίνακα. Ακολούθως, για την προσέγγιση, κρατάμε τα k πρώτα στοιχεία επιτυγχάνοντας τη συμπίεση της εικόνας (εν γένει, τη μείωση των δεδομένων). Έτσι, ο `M`, είναι ένας πίνακας προσέγγισης της εικόνας. Στον νέο πίνακα `M` εφαρμόζουμε τη συνάρτηση `clamp` ($= \text{!}$) ούτως ώστε να διασφαλίζουμε ότι οι τιμές παραμένουν εντός του διαστήματος $[0,1]$. Κατά αυτόν τον τρόπο, τα σφάλματα στρογγύλευσης που προκύπτουν από τον πολλαπλασιασμό πινάκων μεταξύ πραγματικών αριθμών, παραμένουν εντός του διαστήματος $[0,1]$ ώστε να ερμηνευτούν σωστά από την Julia και να αποφευχθούν τα λευκά, από λάθος, φωτισμένα πιξελ (`clipping`).

Τέλος, μέσω της μοσαισιω, επιτυγχάνουμε την παρουσίαση των της αρχικής εικόνας και των συμπιεσμένων εκδοχών αυτής, της μίας δίπλα στην άλλη.

```

# Calculate the SVD of the grayscale image img_gray.
svdfactor = svd(img_gray)

# Create compressed images for different values of k
ks = [10, 50, 100]
imgs = map(ks) do k
    Gray.(rank_approx(svdfactor, k))
end

plot(mosaicview(img_gray, imgs...; nrow=1, npad=10))

```

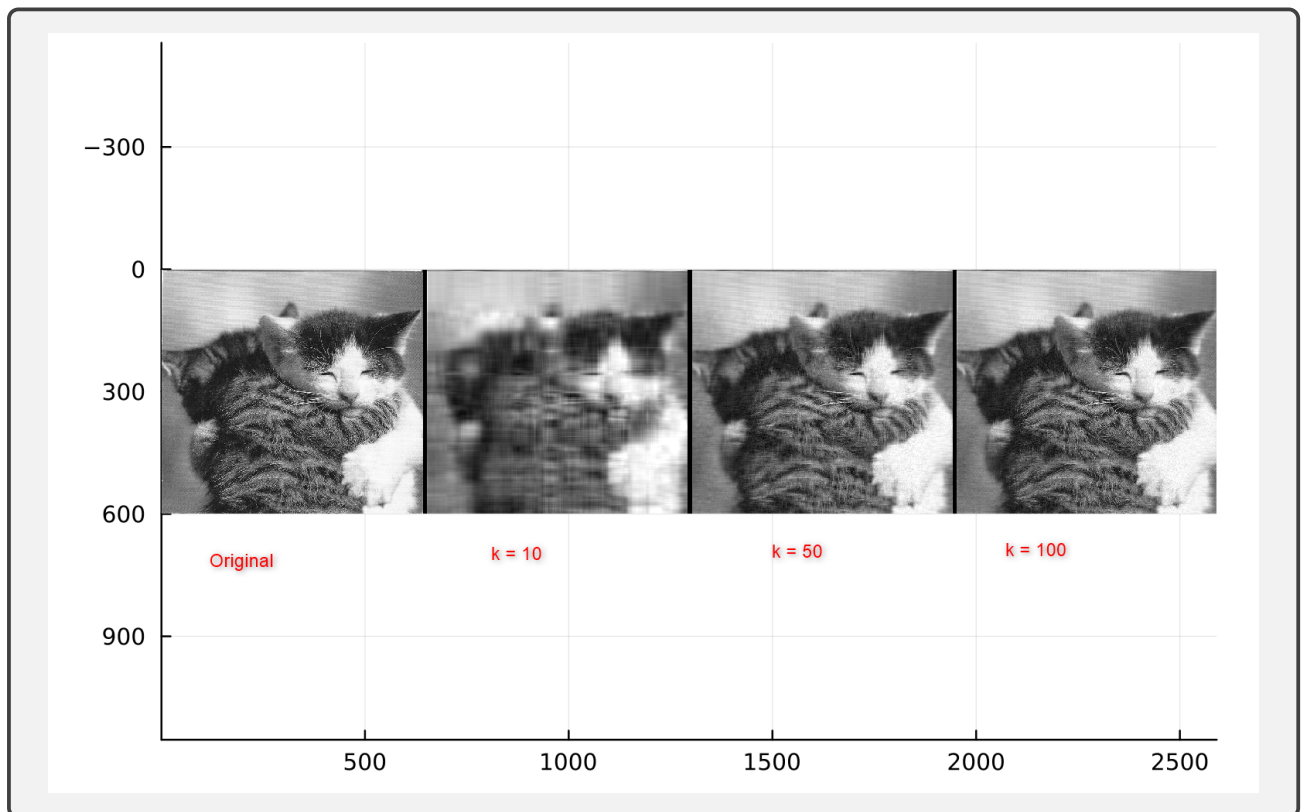


Figure 1: Cats - Gray Image

Παρατηρούμε ότι προσέγγιση για $k = 50$ είναι πολύ καλή

3.2 Έγχρωμη Εικόνα - RGB³

```
using Images, TestImages, LinearAlgebra, Plots

img_path = "C:\\Users\\yanis\\Downloads\\cats_medium.jpg"
img = load(img_path)
img = float.(img)

channels = channelview(img)

function rank_approx(F::SVD, k)
    U, S, V = F
    M = U[:, 1:k] * Diagonal(S[1:k]) * V[:, 1:k]'
    clamp01!(M)
end

svdfactors = svd.(eachslice(channels; dims=1))
imgs = map((10, 50, 100)) do k
    colorview(RGB, rank_approx.(svdfactors, k)...)
end

plot(mosaicview(img, imgs...; nrow=1, npad=10))
```

³RGB (Red, Green, Blue) είναι ένα μοντέλο που χρησιμοποιείται για την αναπαράσταση εικόνων με βάση τα χρώματα: Κόκκινο, Πράσινο και Μπλε. Κάθε pixel σε μια εικόνα RGB αναπαρίσταται από τρεις τιμές, που αντιστοιχούν στις εντάσεις του κόκκινου, του πράσινου και του μπλε. Συνδυάζοντας αυτά τα τρία χρώματα με διάφορους τρόπους, μπορούμε να παράγουμε ένα ευρύ φάσμα χρωμάτων. Για μια έγχρωμη εικόνα, η SVD μπορεί να εφαρμοστεί ξεχωριστά σε κάθε κανάλι χρώματος (R, G, B), δηλαδή εφαρμόζεται ξεχωριστά σε κάθε έναν από αυτούς τους πίνακες. Αυτό έχει ως αποτέλεσμα τρία σύνολα πινάκων U , Σ και V^T .

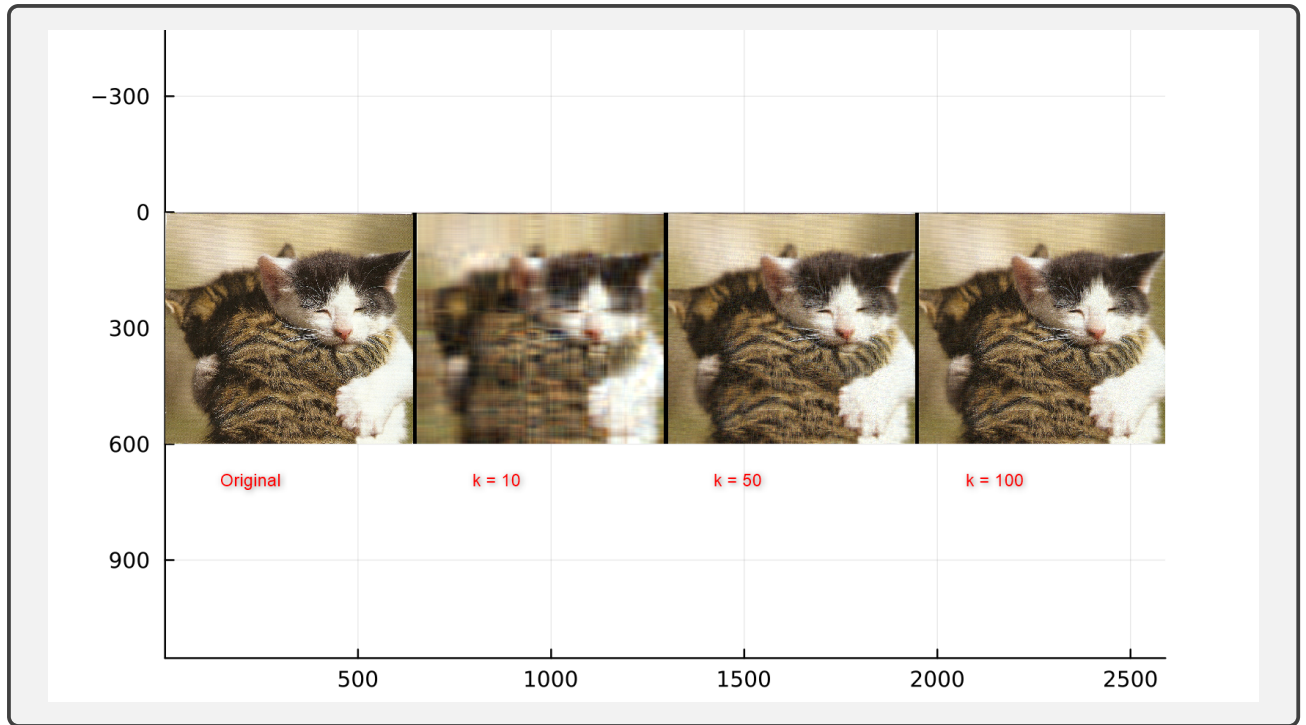


Figure 2: Cats - Coloured Image

4 Επέκταση σε Μεγαλύτερες Διαστάσεις Εικόνas

Ας υποθέσουμε ότι οι διαστάσεις της εικόνας αυξάνονται, δηλαδή m και n γίνονται μεγαλύτερες. Θα εξετάσουμε αν το ποσοστό συμπίεσης παραμένει σταθερό ή αυξάνεται.

Για να απλοποιήσουμε την ανάλυση, ας θεωρήσουμε ότι οι διαστάσεις αυξάνονται με την ίδια αναλογία, δηλαδή $m = n$.

Ο τύπος του ποσοστού συμπίεσης γίνεται:

$$\text{Ποσοστό Συμπίεσης} = \left(\frac{k(2m+1)}{m^2} \right) \times 100\% \approx \left(\frac{2k}{m} \right) \times 100\%$$

Για μεγάλες τιμές του m , ο όρος $\frac{2k}{m}$ γίνεται πολύ μικρός ⁴, πράγμα που σημαίνει ότι το ποσοστό συμπίεσης μειώνεται καθώς m αυξάνεται. Αυτό υποδηλώνει ότι το μέγεθος της συμπίεσμένης εικόνας γίνεται μικρότερο σε σχέση με το αρχικό μέγεθος όταν οι διαστάσεις της εικόνας αυξάνονται.

4.1 Επίδραση του Βαθμού Προσέγγισης k

Ο βαθμός προσέγγισης k παίζει καθοριστικό ρόλο στο ποσοστό συμπίεσης. Συγκεκριμένα:

⁴Όταν το m είναι μεγάλο, ο όρος $\frac{k}{m^2}$ γίνεται πολύ μικρός σε σύγκριση με τον όρο $\frac{2k}{m}$. Συνεπώς, μπορούμε να τον αγνοήσουμε:

$$\frac{2k}{m} + \frac{k}{m^2} \approx \frac{2k}{m}$$

- Σταθερό k : Αν ο βαθμός προσέγγισης k παραμένει σταθερός ενώ οι διαστάσεις m και n αυξάνονται, το ποσοστό συμπίεσης μειώνεται καθώς οι διαστάσεις αυξάνονται. Αυτό συμβαίνει επειδή ο αριθμητής του τύπου $\frac{k(m+n+1)}{m \times n}$ αυξάνεται γραμμικά με m και n , ενώ ο παρονομαστής αυξάνεται τετραγωνικά.
- Μεταβλητό k : Αν ο βαθμός προσέγγισης k αυξάνεται αναλογικά με τις διαστάσεις m και n , δηλαδή $k = \alpha m$ όπου α είναι μια σταθερά αναλογίας, τότε το ποσοστό συμπίεσης γίνεται:

$$\text{Ποσοστό Συμπίεσης} = \left(\frac{\alpha m(2m+1)}{m^2} \right) \times 100\% \approx 2\alpha \times 100\%$$

Σε αυτή την περίπτωση, το ποσοστό συμπίεσης παραμένει σταθερό καθώς οι διαστάσεις αυξάνονται, αφού η αναλογία $\frac{k}{m}$ παραμένει σταθερή.

4.2 Παράδειγμα σε Μεγαλύτερες Διαστάσεις

Ας υποθέσουμε ότι έχουμε μια ασπρόμαυρη εικόνα ανάλυσης $1920 \times 1080 = 2,073,600$ pixels (Full HD) και χρησιμοποιούμε προσέγγιση με $k = 50$.

- **Αρχική Μνήμη:** $1920 \times 1080 = 2,073,600$ pixels
- **Συμπιεσμένη Μνήμη:** $50 \times (1920 + 1080 + 1) = 50 \times 3001 = 150,050$ pixels
- **Ποσοστό Συμπίεσης:** $\left(\frac{150,050}{2,073,600} \right) \times 100\% \approx 7.23\%$

Από το προηγούμενο παράδειγμα για μια εικόνα 640×597 pixels με $k = 50$, το ποσοστό συμπίεσης ήταν περίπου 16.19%. Σε αυτή την περίπτωση, με αυξημένες διαστάσεις, το ποσοστό συμπίεσης μειώθηκε στο 7.23%, υποδηλώνοντας καλύτερη συμπίεση.

4.3 Υλοποίηση με JULIA (Full HD Image)

```
using Images, TestImages, LinearAlgebra, Plots

img_path = "C:\\Users\\yanis\\Downloads\\Orca HD image.jpg"
img = load(img_path)
img = float.(img)

channels = channelview(img)

function rank_approx(F::SVD, k)
    U, S, V = F
    M = U[:, 1:k] * Diagonal(S[1:k]) * V[:, 1:k]'
    clamp01!(M)
end

svdfactors = svd.(eachslice(channels; dims=1))
imgs = map((10, 50, 100)) do k
    colorview(RGB, rank_approx.(svdfactors, k)...)
end

plot(mosaicview(img, imgs...; nrow=1, npad=10))
```

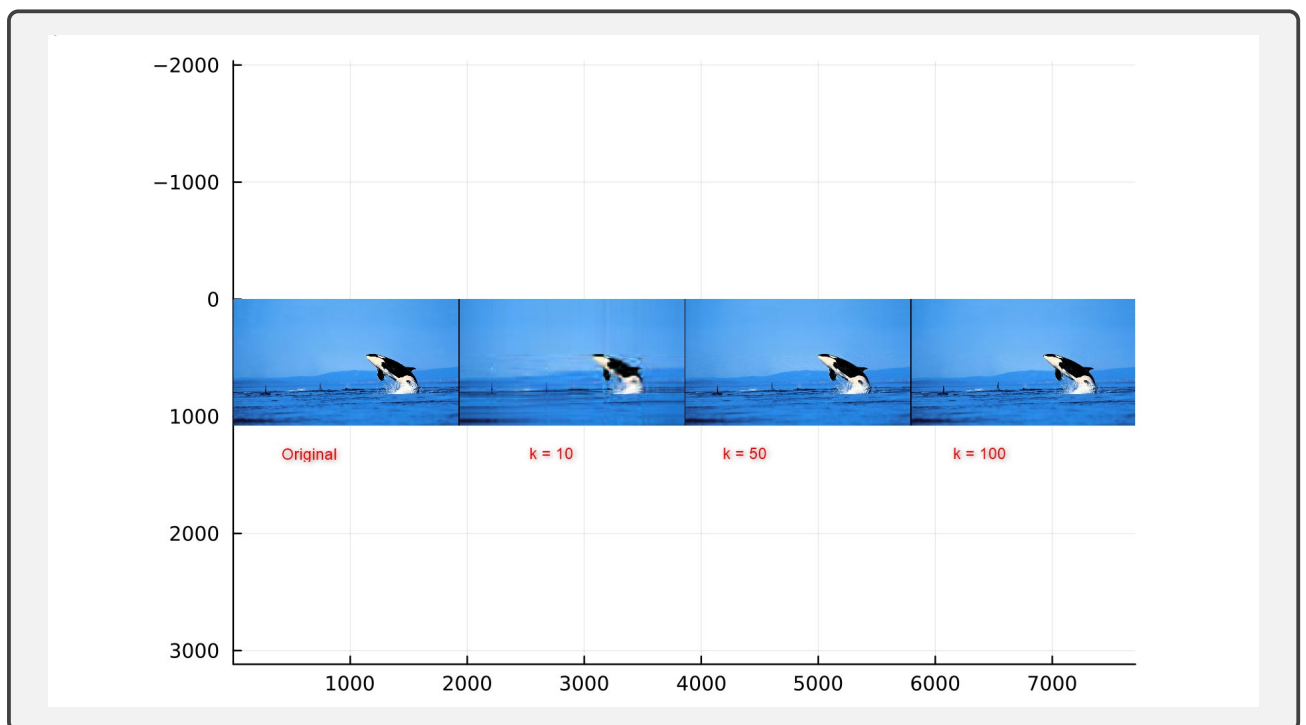


Figure 3: Orca - Coloured Image

5 Συμπέρασμα

Από την προηγούμενη ανάλυση μπορούμε να διαπιστώσουμε την αποτελεσματικότητα της SVD στην συμπίεση εικόνας, ιδιαίτερα όταν χειριζόμαστε μεγάλες διαστάσεις. Η επιλογή του βαθμού προσέγγισης k είναι κρίσιμη για την επίτευξη ισορροπίας μεταξύ της μείωσης του μεγέθους των δεδομένων και της διατήρησης της ποιότητας της εικόνας. Στον ακόλουθο πίνακα, έχουμε συγκεντρωτικά την πληροφορία για τις τρεις περιπτώσεις που αναλύσαμε.

k	Ασπρόμαυρη	Έγχρωμη	Μεγαλύτερες Διαστάσεις (HD)
Αρχική Μνήμη (pixels)	382,080	1,146,240	2,073,600
Συμπιεσμένη Μνήμη (pixels)	61,900	185,700	150,050
Ποσοστό Συμπίεσης	16.19%	16.19%	7.23%

Πίνακας 1: Συγκεντρωτικός Πίνακας για $k = 50$