

---

**SC Project**

**SVD combined with HillClimbing**

**Mohammad amin Roshani**

**610396104**

---

---

**ایده انجام پروژه:** به هنگام استفاده از svd مشاهده شد که در عکس های با تنوع رنگی بالا تشخیص جزییات سخت میشود پس در عکس هایی که به تعداد محدودی رنگ نیاز از میتوان ابتدا رنگ ها را محدود کرده (که خود حجم را کاهش می دهد) و سپس از svd بهره برد.

همچنین کاهش رنگ ها خود باعث تشخیص مواردی مانند نقاط سرطانی سلول و غیره می شوند.

## توضیح الگوریتم:

الگوریتمی که برای پیاده سازی کد استفاده شده هیل کلایمبینگ با گام های متغیر میباشد. بدین صورت که در ابتدا گام های جستجو بزرگ قرار داده شده و در ادامه که نقاط به نتیجه نزدیک تر می شدند گام ها کوچکتر می شد . علت انتخاب متغیر گام ها این بود که گاهی جواب مطلوب آنقدر فاصله داشت که نیاز بود ابتدا با گام های بزرگ به آن نزدیک شد. در ابتدای الگوریتم تعدادی نقطه رندوم از داخل عکس (موجب بهبود بسیار در نتیجه و زمان خواهد شد (انتخاب می شود) مثلا 2000 تا (و از میان آن ها بهترین ها )بر اساس تابع برازش (برای جستجو شروع به کار می کنند. تعداد همسایه ها برابر تعداد رنگ ها  $(3^3 = 27)$  \* است. زیرا هر کدام از مولفه های رنگی 3 حالت سکون، افزایش و کاهش را دارند و برای صرفه زمانی در هر مرحله تنها 1 رنگ تغییر می کند.

با در نظر گرفتن متغیر گام ها مشکل دور بودن رنگ ها از رنگ های بهینه رفع شد اما مشکل جدیدی که به وجود آمد این بود که در گام های بزرگ اولیه برخی از رنگ ها که از همان ابتدا هم خوب بودن تکان نمی خوردند و از دور گردش خارج شده و قابلیت بهبود را از دست می دادند، راه حلی که برای این مشکل ارایه دادم این بود که گام مشخصی را تعیین میکردم که تا قبل از رسیدن به آن هیچ نقطه ای حتی در صورت سکون از دور گردش خارج نمی شد و پس از آن قابلیت خروج را داشتند و تا آن موقع به اندازه کافی بهبود یافته اند.

مشکل دیگری که نیاز به حل داشت این بود که سرعت کاهش گام در اواخر کار بایستی بسیار کم میشد زیرا در بسیاری از نقاط قابلیت گردش بالایی هنوز احساس میشد برای همین پس از رسیدن به نقطه ای که اجازه خروج به نقاط داده میشد سرعت کاهش گام ها هم کاهش میابد. همچنین به علت آوردن بالای الگوریتم برای محاسبه برآزش و پیدا کردن همسایه ها از نسخه فشرده عکس استفاده می کنیم.

در ادامه از ویژگی عکس موجود یعنی تعداد رنگ های کم استفاده کرده و با استفاده از svd به فشرده سازی بیشتر عکس میپردازیم.

## توابع :

- **Make image:** (رنگها را به عنوان ورودی میگیرد و پس از رنگ آمیزی عکس نهایی را به عنوان خروجی میدهد).
- **Fitness:** (رنگها را به عنوان ورودی گرفته و برازش آن را به صورت اقلیدسی بر روی تمام پیکسل های عکس محاسبه می کند).
- **Best\_neighbour:** (رنگها را به عنوان ورودی گرفته و از میان خودش و تمام همسایه هایش بالا ترین فیتنس را به عنوان نقطه جدید خروجی میدهد، به صورتی که بهترین برازش، مشخصات آن) نقطه (و سکون و یا عدم سکون با تعیین میکند).
- **PSNR:** این تابع کامنت شده زیرا تابع cv2.PSNR نتایج دقیق تری میدهد، اما از آن هم می توان استفاده کرد.
- **Show\_images:** برای چاپ عکس
- **Compress\_image:** برای فشرده سازی یک عکس رنگی از جی بی توسط اس وی دی.

## متغیر های مهم :

`n_colors`: تعداد رنگ های مورد استفاده برای رنگ آمیزی.  
`alpha`: گام شروع گردش را تعیین میکند.  
`img_path_input`: مسیر خواندن عکس ورودی.  
`img_path_output`: مسیر ذخیره عکس خروجی.  
`n_iterations`: تعداد گردش های الگوریتم را تعیین میکند.  
`n_initial_points`: تعداد نقاط آغازین (رندوم از داخل عکس).  
`n_points`: تعداد نقاط برگزیده برای شروع جستجو.  
`stop_stuck`: گام گردشی که پس از آن اجازه خروج نقاط ساکن صادر میشود.  
`alpha_dec`: ضریب کاهش گام شروع گردش. (`alpha`)  
`points`: دیکشنری نقاط الگوریتم به صورت {رنگ: برازش}.  
`final_points`: نقاط پس از اتمام ادوار گردش به صورت دیکشنری.  
`rr,rg,rb`: ماتریس ساخته شده از `svd` آر، جی و بی.

## نقاط قوت الگوریتم:

- قدرت گردش بالا و عدم قرار گیری در بهینه موضعی
- فیتنس های بسیار بالا به علت بهینه سازی گام های گردش
- گردش خوب به علت تعداد بالای همسایه ها
- نتایج بسیار دقیق به علت انتخاب رنگ های اولیه از داخل عکس
- کاهش حجم بسیار بالا با تغییر بسیار کم کیفیت تصویر
- قابلیت استفاده در داده های بزرگ با بخش بندی واضح

## نقاط ضعف الگوریتم:

- زمان نسبتا بالای الگوریتم (حدود 8 دقیقه به طور میانگین برای 3, 4 و 5 رنگ)

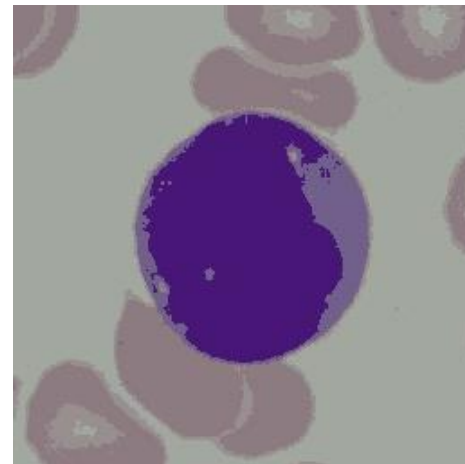
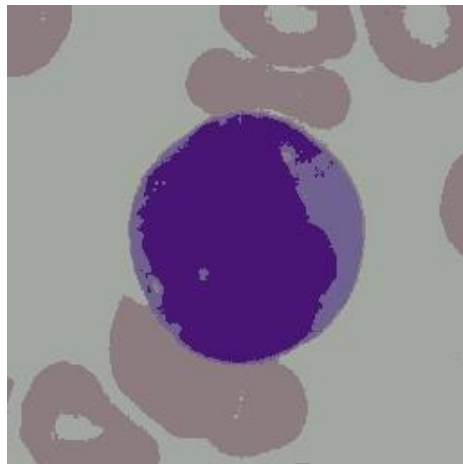
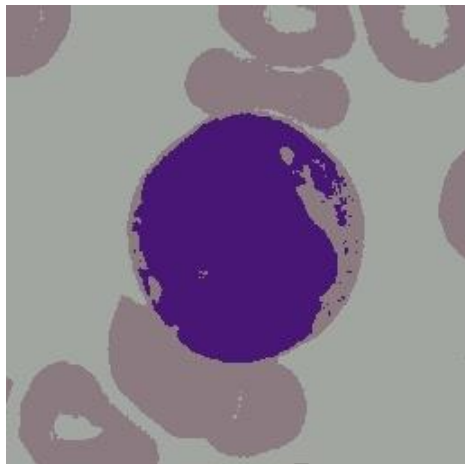
## بررسی بهترین و بدترین نتایج:

در نتایج ما عکس 006 با برازش 9.0436 و عکس 007 با برازش 16.0471 هر دو در 3 رنگ (به ترتیب بهترین و بدترین نتیجه هستند).

با بررسی آن ها متوجه میشویم که الگوریتم به صورت بسیار منطقی عمل کرده و به ازای هر عکس هر چه تعداد سطوح رنگی بیشتر باشد، برازش نیز بیشتر است. بدین معنا که انتخاب تعداد محدود رنگ هر چند بهینه برای رنگ آمیزی آن خطای بالایی میدهد.



img001



Colors

```
[[117.19007576 22.63611031 72.39650773]  
[161.18360402 166.19242403 162.93632346]  
[129.20422291 123.46271389 139.70035282]]
```

```
[[144.36252978 97.50268406 111. ]  
[127.01306749 124.3474122 140.3474122]  
[117.42546986 21. 72.90291428]  
[161. 167. 162. ]]
```

```
[[143.23774733 144.94254349 152.90094789]  
[119.60000901 22. 72. ]  
[160. 167. 162. ]  
[128.93953617 123. 140. ]  
[140.1558242 96. 114.74193267]]
```

Fitness (pp)

10.776908808066891

9.444770548024915

8.822344178653832

PSNR

25.97473973982959

26.90834888358835

27.45651456083154

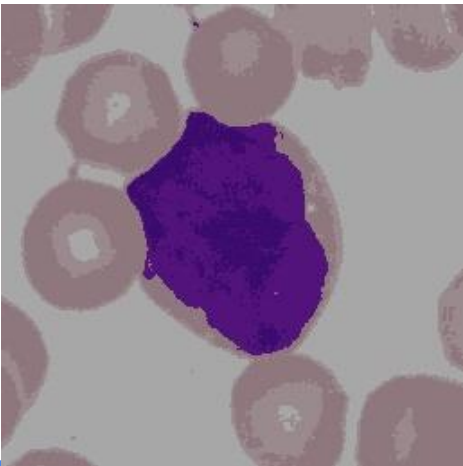
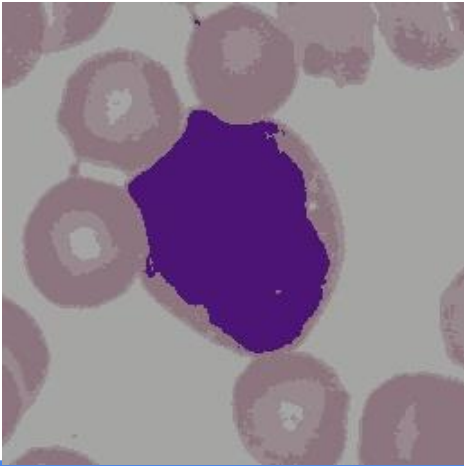
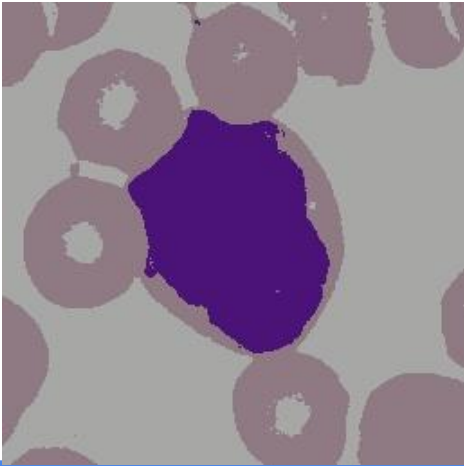
Time (min)

7.502982529004415

9.446418865521748

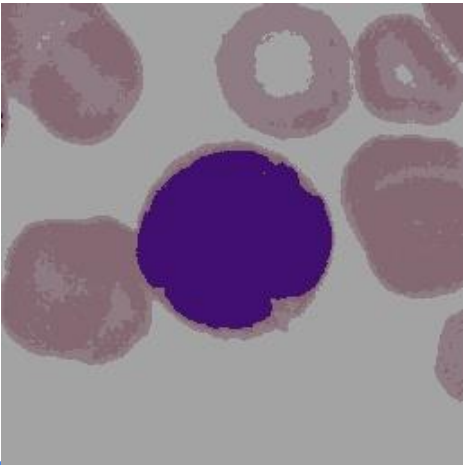
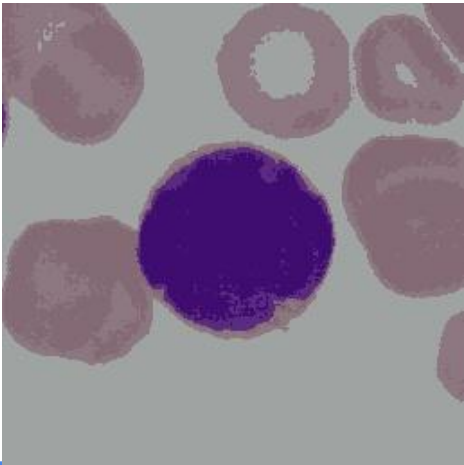
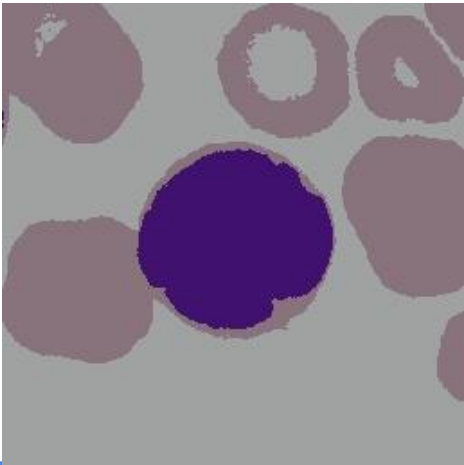
10.183902855714162

img002



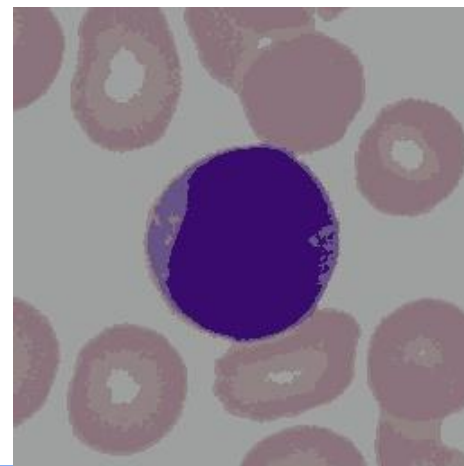
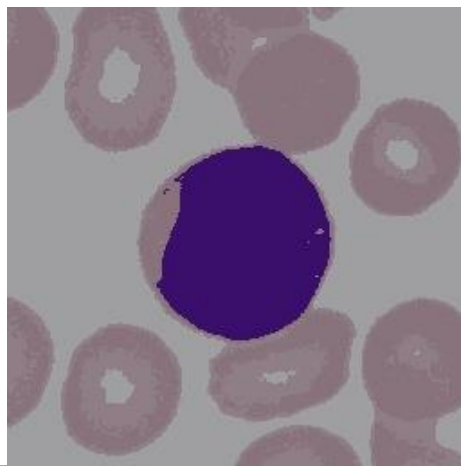
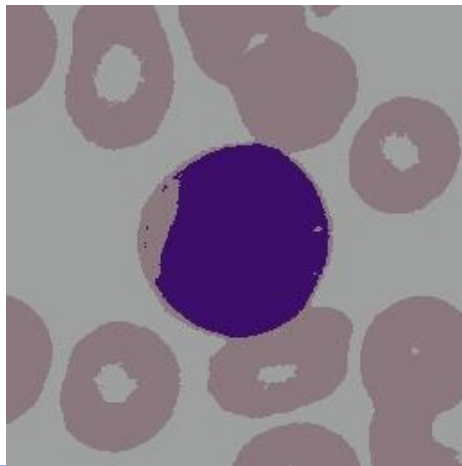
Colors	[[165.90797656 167.01545533 167.06279329] [130.82538898 121.73764185 143.640676 ] [119.26736582 19.82133549 75.8107761]]	[[164.40518491 165.66100313 165.66100313] [126.5159547 117.16795827 140.53002921] [143. 137.48302937 152.48302937] [117.6892031 20. 75. ]]	[[141.15556012 137.35734317 154. ] [125.71594573 120. 142. ] [110.254725 11. 60.76277399] [122. 19.9742375 82. ] [167. 167. 167. ]]
Fitness (pp)	11.695370549948773	10.053562700325081	9.469696220655441
PSNR	24.20295178150863	25.16312024429682	26.48345301057405
Time (min)	6.026001453399658	6.976185131072998	9.152117590109507

img003



Colors	<pre>[[123.96639162 114.54303894 136.88979206] [112.28193056 17.07808393 65.35858735] [161.70037352 161.95370061 160.82225579]]</pre>	<pre>[[162.48462699 163. 162. ] [132.60348135 123.60348135 143.60348135] [114.6990929 105. 132. ] [111.20811482 15.44006171 65.85779933]]</pre>	<pre>[[162. 164. 159.53546425] [116.9791205 106.64288407 131. ] [129.98671105 43.85455847 90.34960702] [129. 120. 141. ] [112.25738824 13. 63.54066658]]</pre>
Fitness (pp)	10.69984801058484	9.18410522830858	9.055969449184884
PSNR	25.37375250666375	26.70360082307472	26.97959157884431
Time (min)	7.14731407960256	7.621532622973124	8.876194655895233

img004



Colors

[[160.3493157 160.00969924 158.87150621]  
[127.22054544 120.12540095 139.06605524]  
[106.36818426 14.1118438 60.10408782]]

[[123.55720534 114. 136.60037439]  
[161.22930399 159. 159. ]  
[134.35659248 126.63265781 144.21147299]  
[107.95634501 14. 58.3777964]]

[[131.27093291 131. 145. ]  
[134.42795 84.988605 103.90345 ]  
[159. 160. 158. ]  
[125. 115. 138. ]  
[109. 11. 57. ]]

Fitness (pp)

10.33195287461982

9.029098187888806

8.557398658418439

PSNR

27.58989310940424

28.26596529075371

29.35924597104929

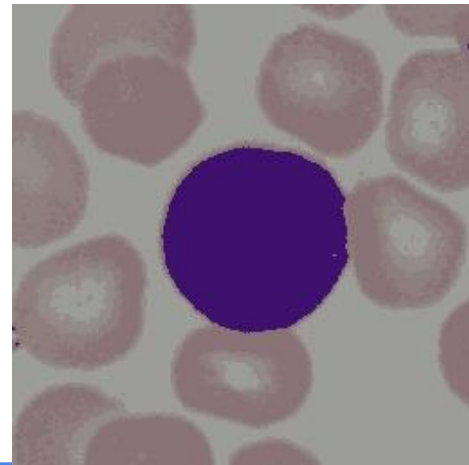
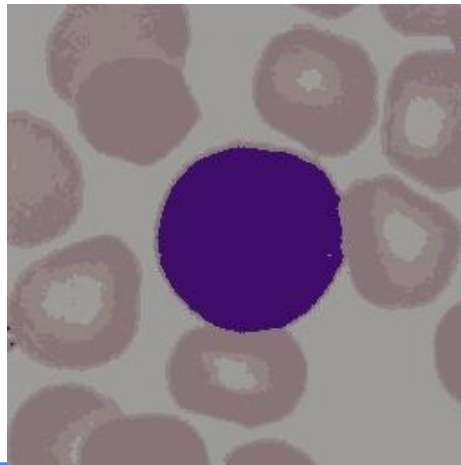
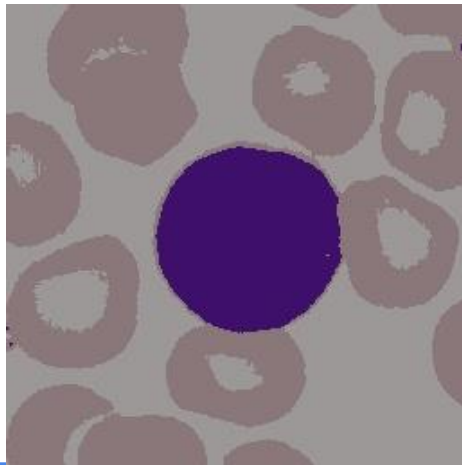
Time (min)

5.695690854390462

6.216168165206909

9.306481432914733

img005



Colors

```
[[108.99946076 15.52477703 63.61165856]  
[151.96744843 152.72597685 155.24226716]  
[121.42440455 119.6112347 138.57462068]]
```

```
[[121. 117.36369495 136.53740083]  
[107.50376927 14.45507292 63.87735837]  
[132.37579056 132.37579056 147. ]  
[153. 155. 156. ]]
```

```
[[119.18119846 115.7083009 138.32818167]  
[126.50664093 126.0257625 142.50664093]  
[153. 156.32744951 156. ]  
[137.31372038 139.31579327 148. ]  
[109. 17. 61.62132958]]
```

Fitness (pp)

10.597367561369337

8.976727102402068

8.351125606223805

PSNR

26.20896557477546

27.91157110113545

28.38748408557639

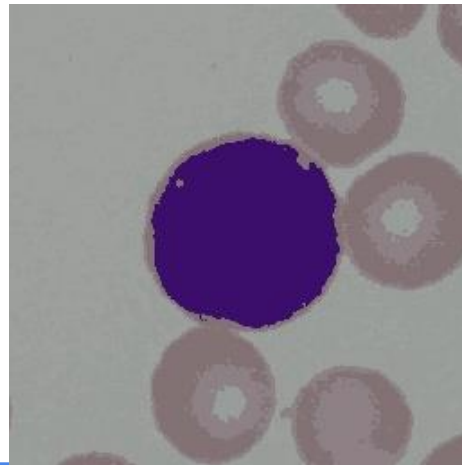
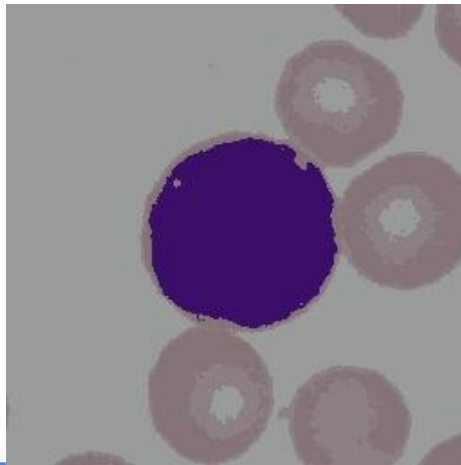
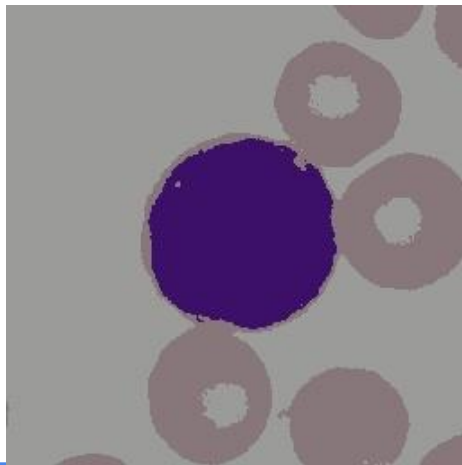
Time (min)

6.725853097438812

7.437700355052948

11.07057458559672

img006



Colors

```
[[122.56161585 118.89758915 135.48832439]  
[154.26026551 156.86530229 155.77345769]  
[105.55035289 15.80282671 60.63359382]]
```

```
[[105.    14.89973487 59.42225529]  
[119.51697063 115.35073498 133.   ]  
[155.    156.57380545 154.82346383]  
[132.90516999 128.72521212 141.   ]]
```

```
[[133.6499851 129.50259182 142.73157909]  
[153.    157.31579327 153.   ]  
[107.30699608 14.44405156 59.27817675]  
[116.62220725 114.    131.06109201]  
[157.    160.    158.   ]]
```

Fitness (pp)

9.043612773308277

8.166031553483297

7.6654269984021735

PSNR

28.49024063580349

29.46279429522818

30.47159914192822

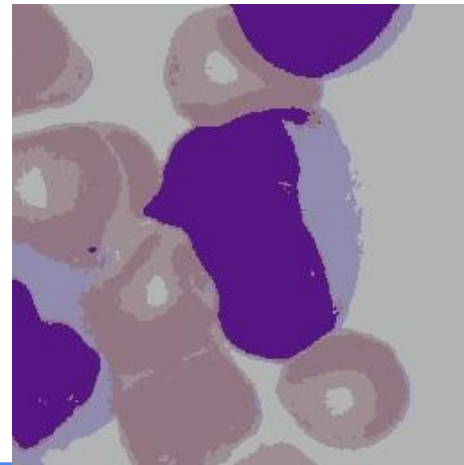
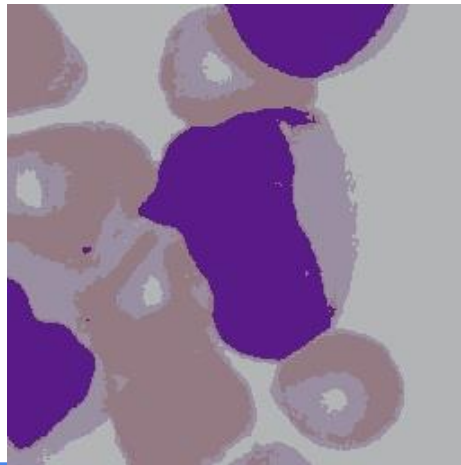
Time (min)

7.173699676990509

8.000317911307016

8.509229183197021

img007



Colors

[[137.58363093 125.33895947 150.12894999]  
[179.74365129 177.86055308 177.88855732]  
[134.53751936 25.84523095 89.34283027]]

[[131. 122. 148. ]  
[182.01700553 180.37682556 179.93547364]  
[136.33849156 27.47043577 89.86564886]  
[162.6287935 142.51971574 153.37038802]]

[[132.49661237 118. 147. ]  
[175.41251047 139. 144. ]  
[147.07467766 142.88674752 159. ]  
[181. 179. 178. ]  
[132.1864231 22.97833855 86.1864231]]

Fitness (pp)

16.047128013619226

13.528098315773175

12.587268468423343

PSNR

20.37724127712698

23.42715256458031

24.79287357464852

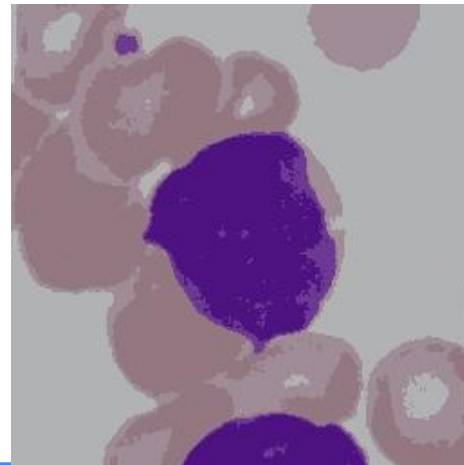
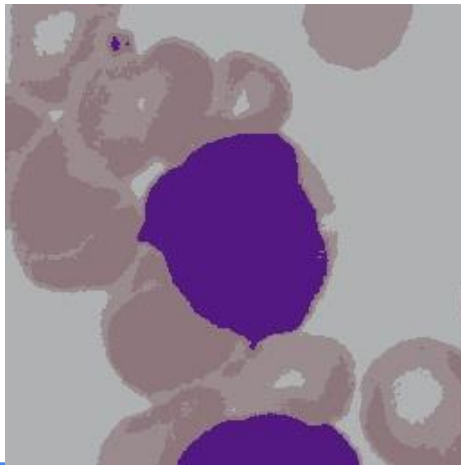
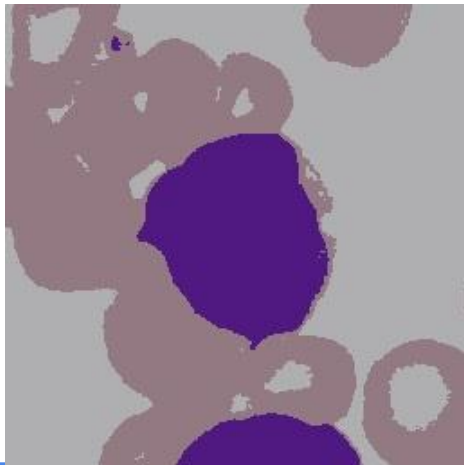
Time (min)

5.22511016925176

6.748797802130381

9.639236807823181

img008



Colors

```
[[178.67502017 175.7771419 175.85689333]  
[129.27447752 24.182075 80.72762095]  
[130.87821756 122.26907769 147.93349475]]
```

```
[[129. 24. 83.23564528]  
[124. 118.15071698 141.94133426]  
[142.89083921 140.16746331 154.49659063]  
[179. 177. 176. ]]
```

```
[[151. 140. 160. ]  
[149.59847716 57.59847716 108.00287996]  
[129. 120. 147. ]  
[129. 19. 78. ]  
[180. 178. 177. ]]
```

Fitness (pp)

14.286955470729337

11.74208506173433

10.878644512864748

PSNR

21.39986906972702

24.95458178191465

25.12823600534301

Time (min)

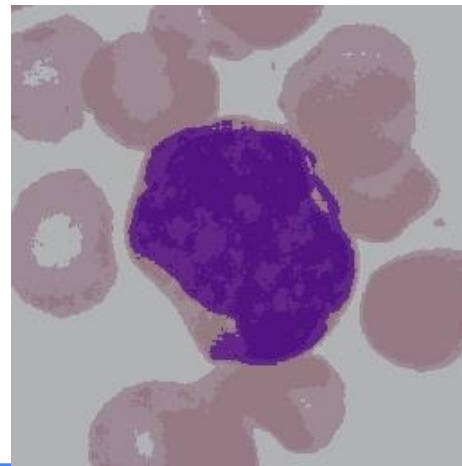
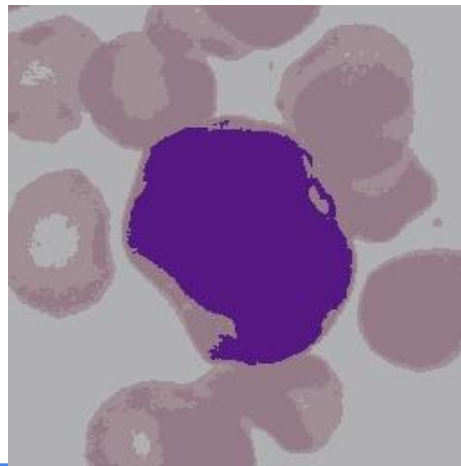
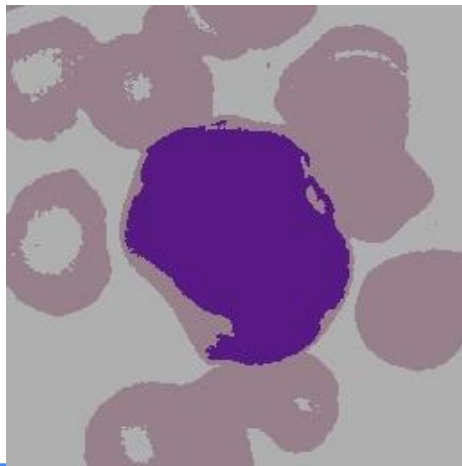
5.31704181432724

5.4144727945327755

8.286703701814016



img009



Colors

[[140.17560261 128.03699817 151.87967467]  
[176.64421492 175.30007546 175.93095106]  
[131.73155149 25.71763357 87.30391448]]

[[148.16832802 142.56561076 160.5983785]  
[135. 123.33489275 149.33489275]  
[130.05933422 24.71329842 86.17322356]  
[178. 176. 176. ]]

[[180. 178. 177. ]  
[131. 122. 149. ]  
[129. 20. 82. ]  
[149. 138. 158. ]  
[137. 38.82938392 99. ]]

Fitness (pp)

13.817759126653872

11.141867700160676

10.525901296802893

PSNR

31.86992589678724

33.4073715162413

35.98471383561312

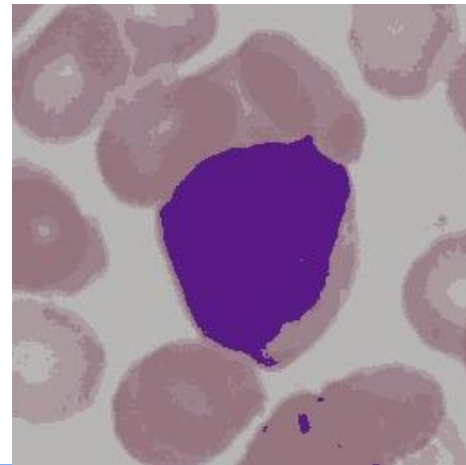
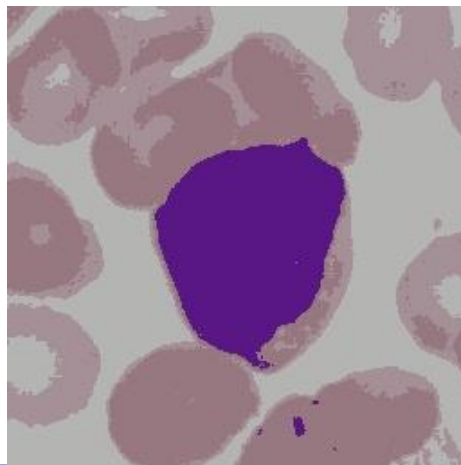
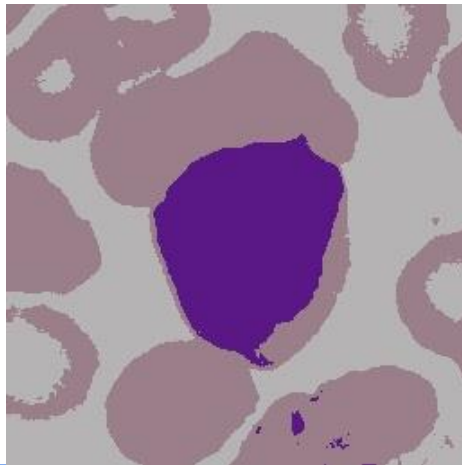
Time (min)

7.29641858736674

8.464795060952504

8.5146426598231

img010



Colors

```
[[137.29635566 126.61982563 154.95994464]  
[133.68430317 24.78966737 89.55267694]  
[180.4500918 179.16009016 181.74216975]]
```

```
[[151.41680818 144.91750038 165. ]  
[179.57557883 180. 180.54405587]  
[134.36153725 23.44006171 89. ]  
[129.73365749 120. 151. ]]
```

```
[[127. 118.5416364 149.91499901]  
[133. 26.45619616 88. ]  
[181. 182.17323733 183. ]  
[158.4172287 155.14637776 171.06766482]  
[141. 130. 156. ]]
```

Fitness (pp)

15.993253592808642

12.528372365041637

11.543654010602323

PSNR

20.53121707507172

23.56363695366555

24.33387171222647

Time (min)

6.295353106657664

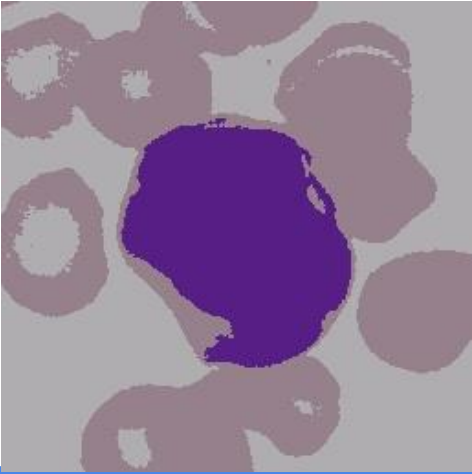
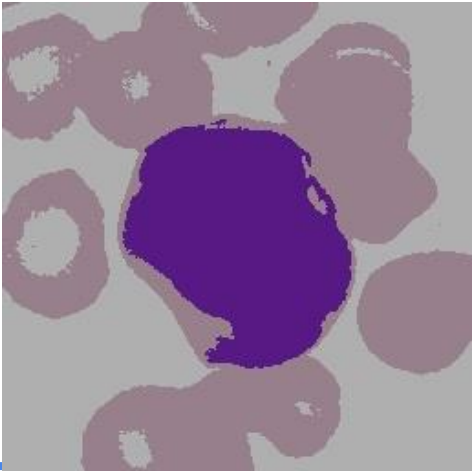
7.026523427168528

10.89017915725708

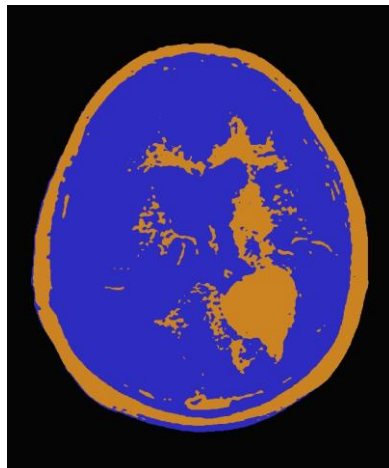
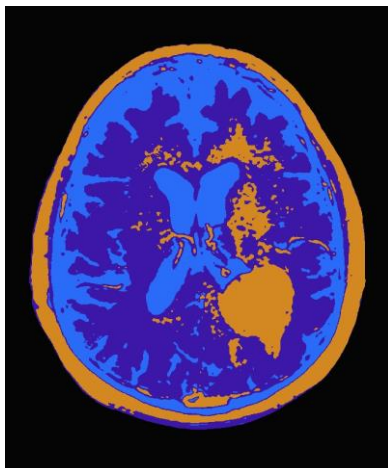
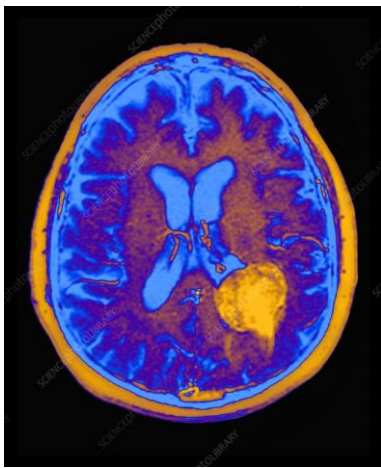
## مقایسه نتایج با مقاله:

همانطور که مشاهده میشود در عکس ۱۰۰۹ الگوریتم ما از مقاله فیتنس بهتر و PSNR بدتری داده است.  
زمان مقاله هم مطابق گزارش بسیار بهتر است.

Comparison  
Img009  
Left: my result  
Right: paper  
result



Colors	[[140.17560261 128.03699817 151.87967467] [176.64421492 175.30007546 175.93095106] [131.73155149 25.71763357 87.30391448]]	[[140 128 150 ] [175 173 175 ] [133 31 87  ]]
Fitness (pp)	13.817759126653872	14.6596179889018
PSNR	28.155306556846913	28.3576
Time (min)	7.29641858736674	-



Colors

```
[[131. 122. 148. ]  
[182.01700553 180.37682556 179.93547364]  
[136.33849156 27.47043577 89.86564886]  
[162.6287935 142.51971574 153.37038802]]
```

```
[[137.29635566 126.61982563 154.95994464]  
[133.68430317 24.78966737 89.55267694]  
[180.4500918 179.16009016 181.74216975]]
```

Fitness (pp)

29.360842851244005

32.6596179889018

PSNR

22.78209367977012

27.3576839843984

Time (min)

7.29641858736674

-

## کاربرد:

مشخصا استفاده این پروژه در موارد پزشکی و تشخیص سلول های سرطانی بوده و مراکزی که نیاز به ذخیره سازی عکس ها در اسکیل بسیار بالا دارند و برای تشخیص عکس ها تنها نیازی به تعداد محدودی رنگ دارند میتوانند از این مورد بهره برده و svd نیز به پایین آوردن حجم و ذخیره عکس ها در تعداد بالا کمک خواهد کرد.

## نحوه اجرای کد

آدرس ورودی و خروجی عکس را به ترتیب در `gimg_path_input` و `img_path_output` قرار دهید.

همچنین کانفیگ مناسب هر تعداد رنگ را میتوانید در `config.txt` ببینید. همچنین مستقل میتوانید فشرده سازی عکس را مستقل از الگوریتم هیل کلایمبینگ ران کنید. (فراموش نکنید که سلول اولیه را همواره اجرا کنید.)

- Singular Value Decomposition: Compression of Color Images - by Bethany Adams and Nina Magnoni page 16-26
- Acute lymphoblastic leukemia image segmentation driven by stochastic fractal search – by Krishna Gopal Dhal & Jorge Gálvez<sup>2</sup> & Swarnajit Ray & Arunita Das & Sanjoy Das
- A New Hillclimber for Classifier Systems – by Kwok Ching Tsui and Mark D. Plumbley