

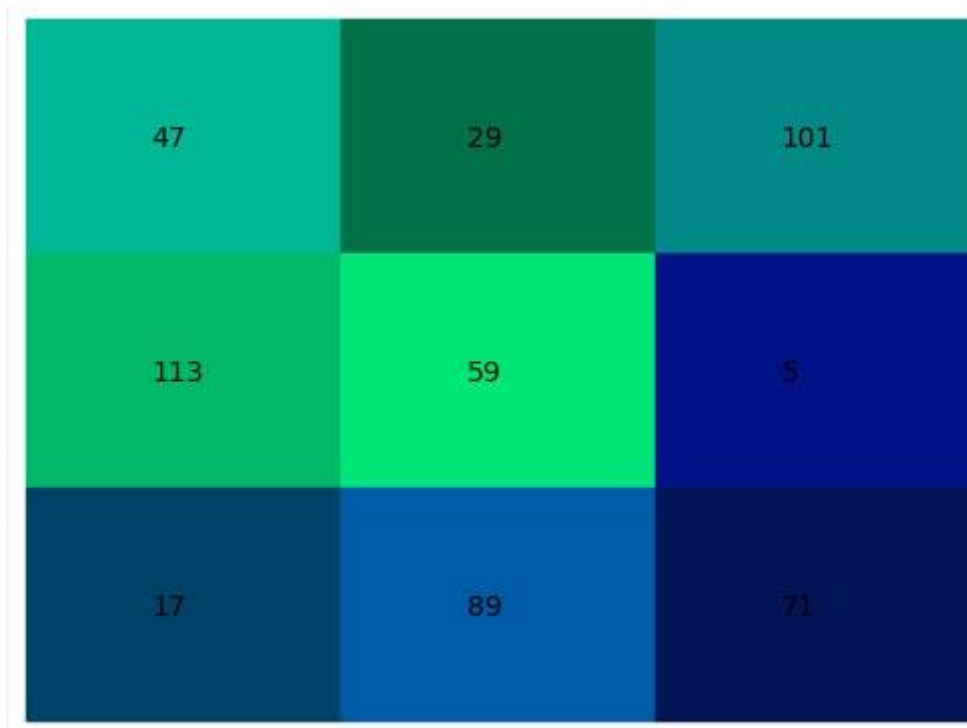
TRADUCCIÓN DE IMÁGENES A COLORES

Es difícil hacer una transformación exacta de los números naturales al formato de color RGB, ya que estamos pasando de un espacio 1D a uno 3D. Optamos por tomar la decisión más simple: el formato de color RGB está compuesto por 3 números correspondientes a cada uno de los colores básicos, pero al final los tres números forman uno solo de la forma #000000.

Por tanto, la transformación consistiría en hacer el paso de decimal a hexadecimal (que es la base que utiliza el RGB). Por ejemplo, el número 5003 sería el color #00138B.

Sin embargo, un posible problema que predecimos en estas imágenes es que los colores son demasiado similares. Para evitar esto, multiplicamos el número por un factor antes de aplicar la transformación. Por ejemplo, si usamos el factor 1000, el 5003 pasa a ser el 5003000, que en RGB sería #4C56F8

Aplicando esta transformación vemos que las matrices tienen colores claramente más variados.





Esta transformación la hacemos con los programas: `pasar_matrices_de_linea_a_colores.py` y `recortar_imagenes.py`

ALGORITMO Y ENTRENAMIENTOS

Utilizamos este programa de *Image Inpainting* (Completado de imágenes) para nuestra prueba del completado de las matrices con colores:

https://github.com/JiahuiYu/generative_inpainting

La idea es que, al entrenar, las máscaras cubran completamente uno de los cuadrados para que no pueda reconstruirlo a partir de un color, que ya tenga. Por ejemplo:

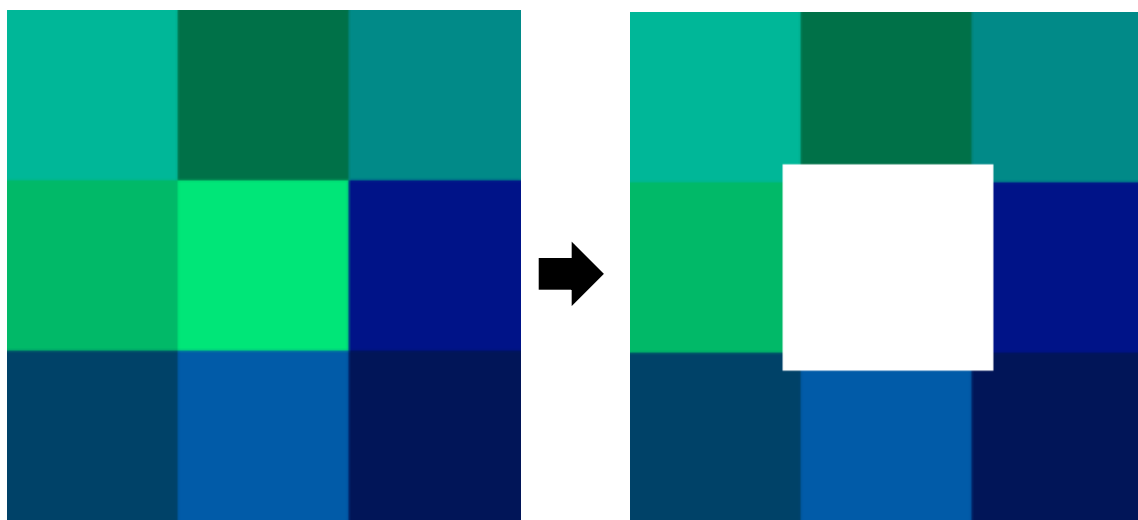


Imagen original

Imagen con máscara

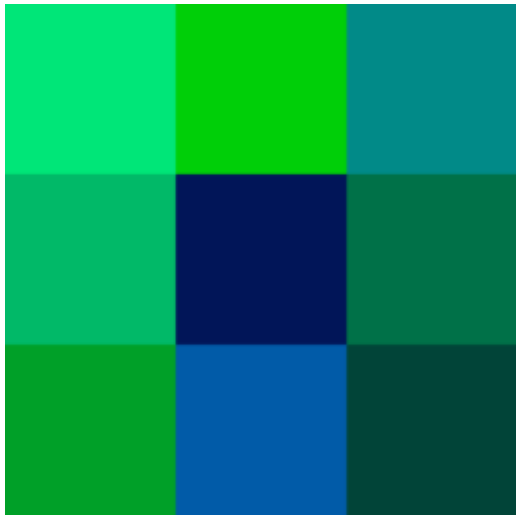
En nuestra prueba usamos simplemente máscaras para el elemento central por temas de sencillez.

PRIMER ENTRENAMIENTO

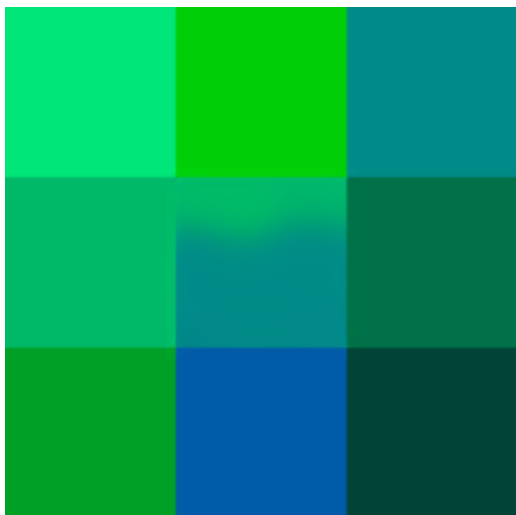
Después de dejar entrenando el modelo *Image inpaiting* un fin de semana, obtuvimos los siguientes resultados. A continuación, ponemos las imágenes de algún ejemplo sacado. Ponemos también una tabla al final con los ejemplos que realizamos.

Ejemplo 1. Matriz del número 71

La matriz original sería:

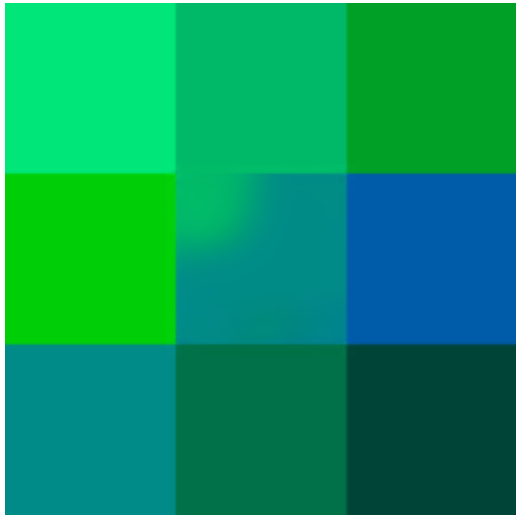


Y la matriz que saca la red neuronal:

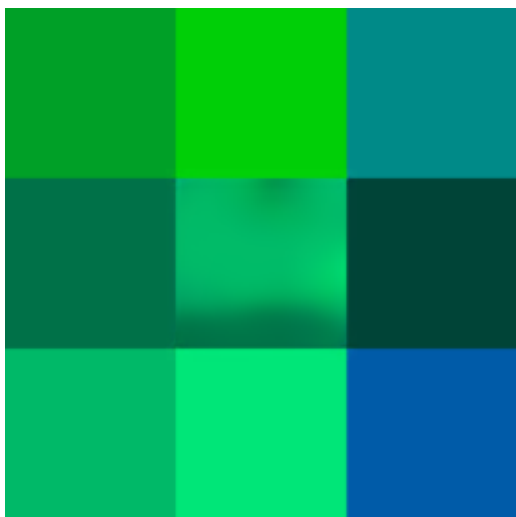


Llegamos a la conclusión de que este resultado no tenía por qué ser erróneo, sino que puede ser que la red neuronal tuviese una función que relacionase este color central de la matriz de output con el número o color correcto central. Es decir, que, para la red neuronal, esta mezcla de colores que vemos en el cuadrado central quizás si signifique el valor central correcto.

Por ello, eprobamos ponerle a la red neuronal la misma matriz del número 71 pero **inversa**, como imagen de input, a ver si sacaba el mismo color que con la matriz original. Esto es lo que hemos hecho y el resultado es el siguiente:



El color central es bastante parecido, lo cual refuerza la teoría expuesta anteriormente. Además, introduciendo la matriz del 71 pero con los cuadrados cambiados de manera aleatoria, es decir, que ya no va a cumplir las reglas de las sumas (la matriz original y la matriz inversa si seguían dichas reglas), obtuvimos lo siguiente:



Observamos que el color central es bastante diferente a los dos anteriores, cosa que sigue reforzando la teoría anterior, ya que el central ha de ser diferente a los anteriores por no seguir la matriz las reglas de las matrices de primos.

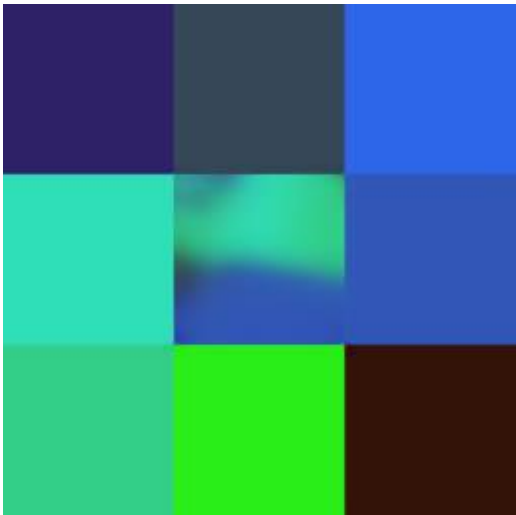
Ejemplo 2. Matriz del número 3251

En este ejemplo, ocurre algo parecido que con el ejemplo primero.

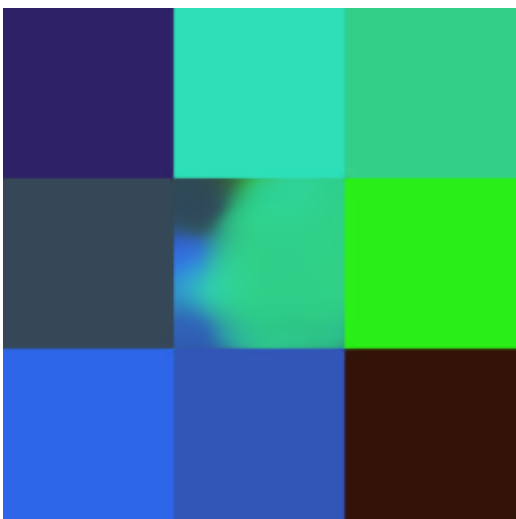
Matriz original:



Matriz sacada por la red neuronal de la matriz original:



Matriz sacada por la red neuronal de la matriz inversa:



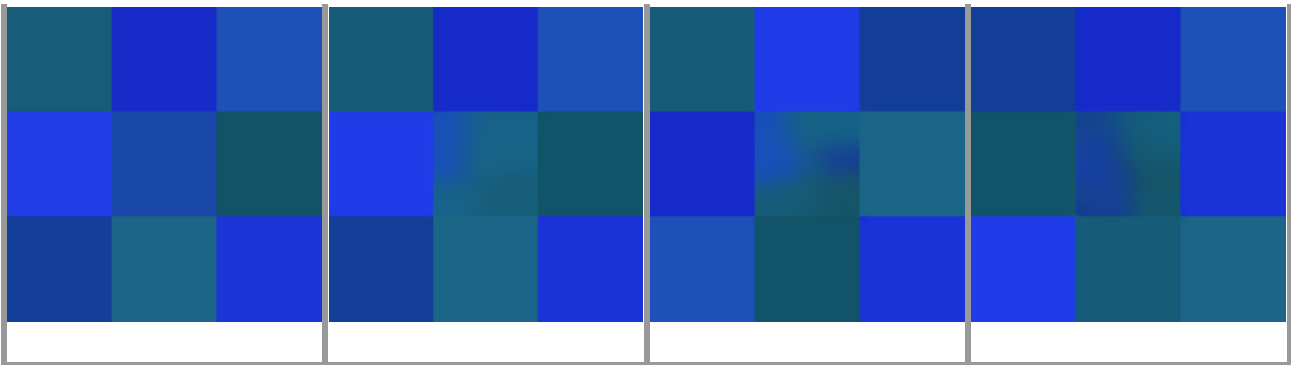
Matriz sacada por la red neuronal de la matriz aleatoria (no cumple las reglas):



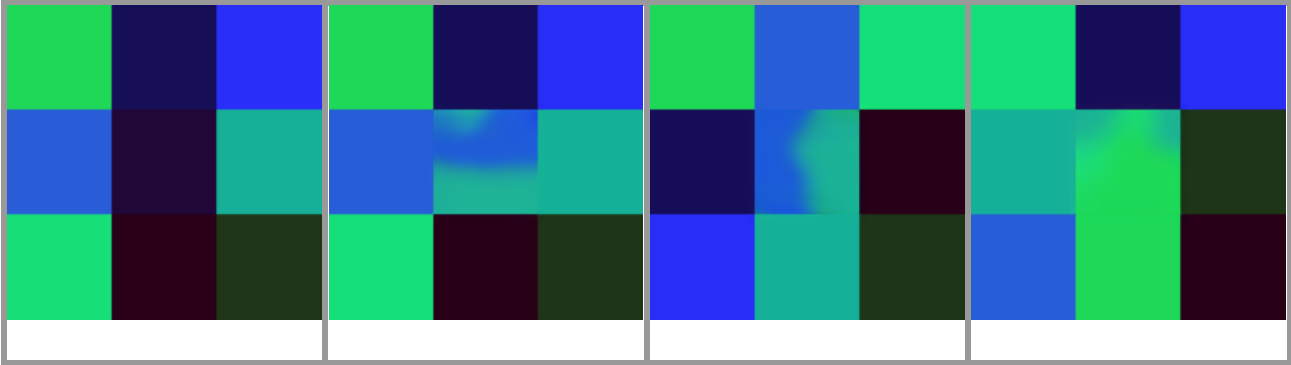
Como podemos ver, también se cumple en este ejemplo que los colores centrales de las matrices originales e inversa sacadas por la red neuronal, son bastante parecidos, en cambio el color obtenido por la red para la matriz aleatoria es muy diferente.

TABLA EJEMPLOS MATRICES

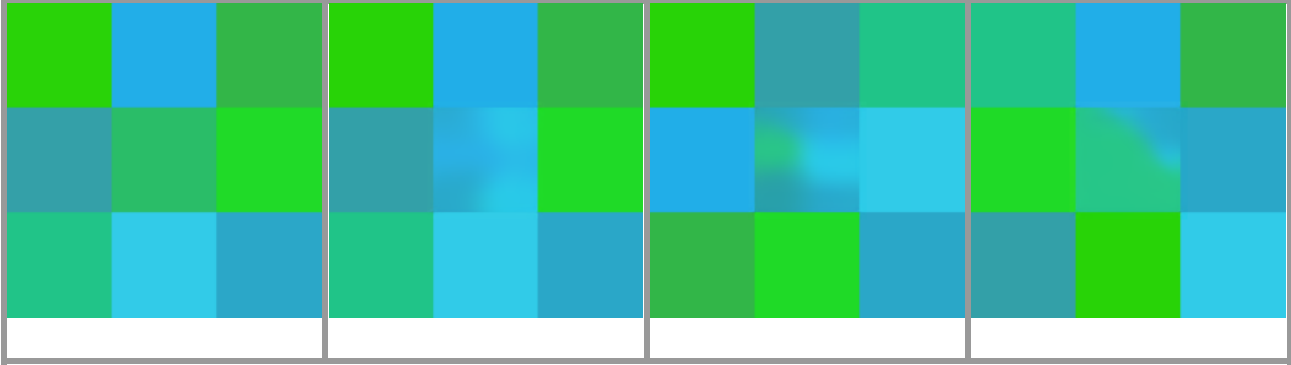
Original	Normal	Inversa	Aleatoria
71			
547			
1657			



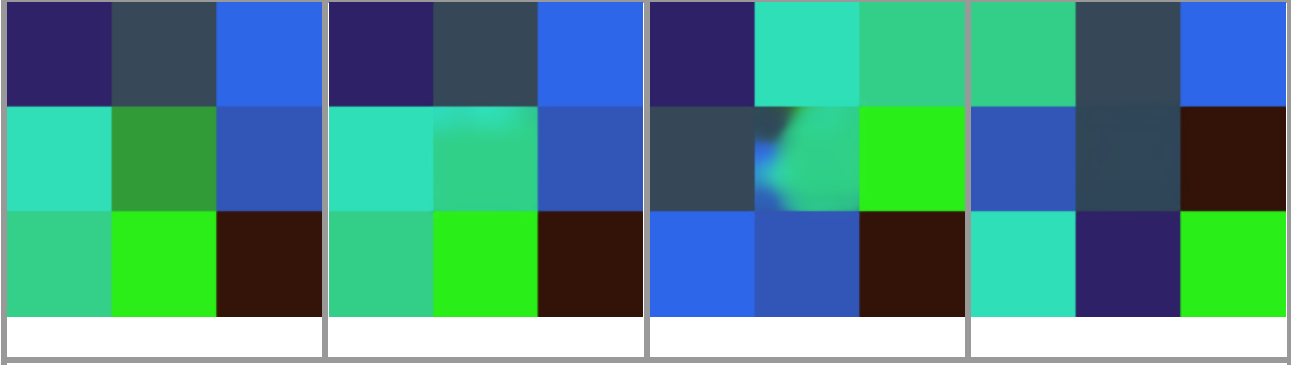
2099



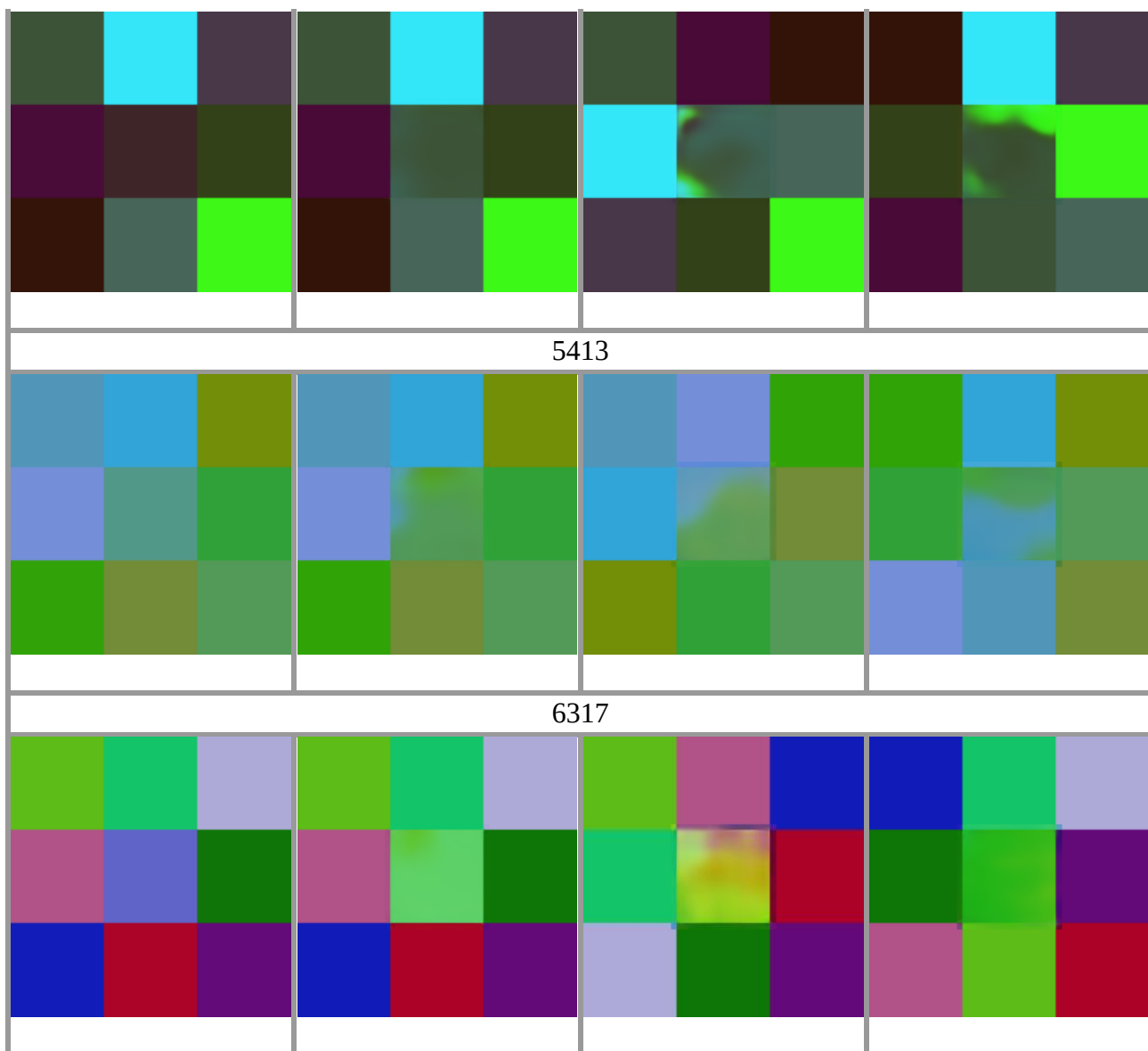
2801



3251



4073



Cabe destacar también que, en la última matriz de ejemplo, la de 6317, la red sí que añadió colores centrales nuevos, que no estaban en la matriz de input.

SEGUNDO ENTRENAMIENTO

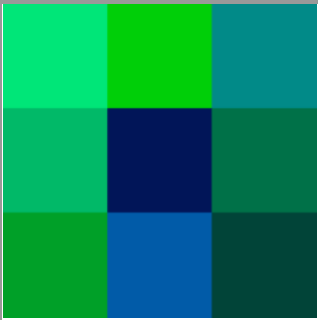
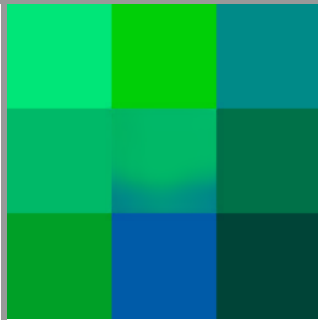
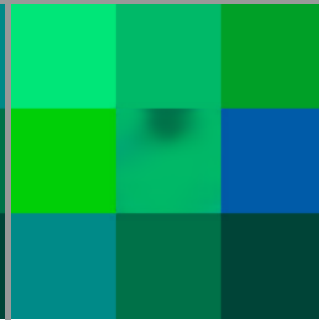
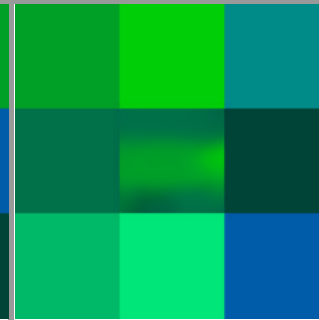
Repetimos la prueba (obtener el elemento central a partir del resto para la matriz original, la inversa y con los colores desordenados de manera aleatoria) pero dejando más tiempo al programa para que se entrene. No vimos mucha mejora respecto de la primera prueba. Ninguna matriz obtuvo colores nuevos a excepción de la última.

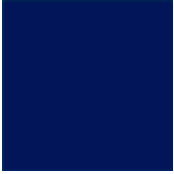

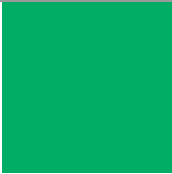
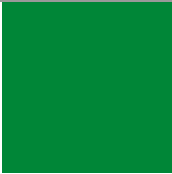
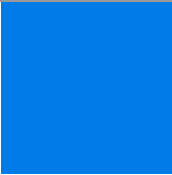
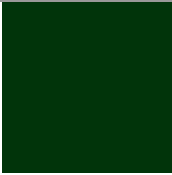



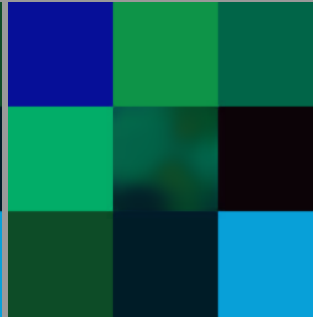
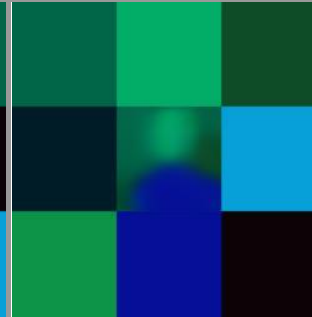


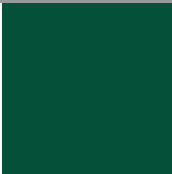
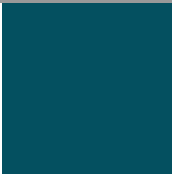

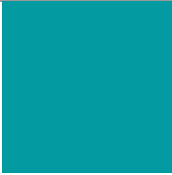
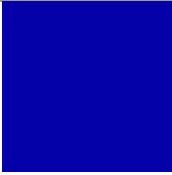
Además, dado que, en la mayoría de ocasiones el cuadrado central no es uniforme, tomamos la mezcla del color resultante de todos los colores contenidos en el cuadrado para poder compararlo mejor con el original. Hemos realizado la mezcla de dos maneras distintas:

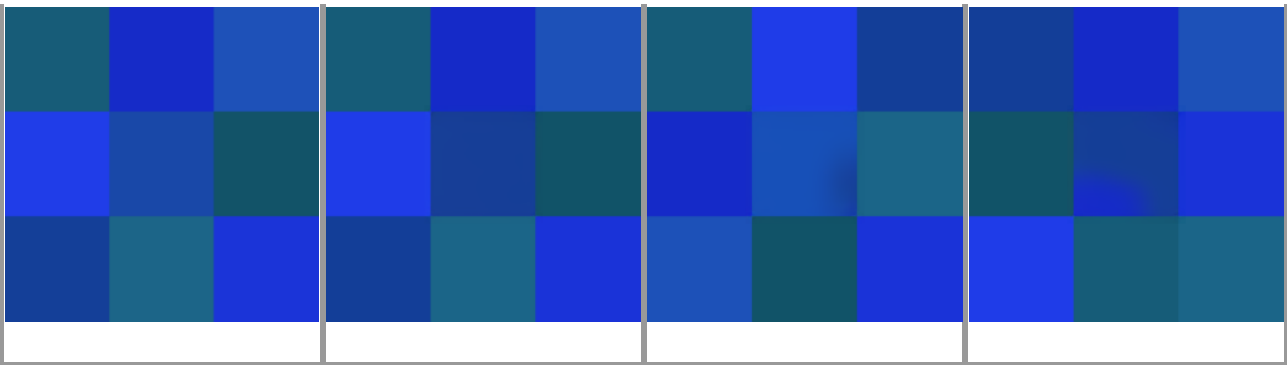
1. Mezcla RGB: Obtenemos los valores de rojo (R), verde (G) y azul (B) de cada píxel y calculamos el valor medio de todos ellos por separado. Los tres números resultantes nos darán el código RGB del color mezcla.
2. Mezcla hexadecimal: obtenemos el código hexadecimal de cada píxel (que tiene la forma #000000) donde los dos primeros números representan el rojo en hexadecimal; los dos siguientes, el verde, y los dos últimos, el azul. Después pasamos este número a decimal y hacemos la media de este número resultante entre todos los píxeles. Finalmente, el número que nos salga, lo volvemos a pasar a RGB.

Estas dos maneras de sumar nos dan colores mezcla notablemente distintos entre sí.

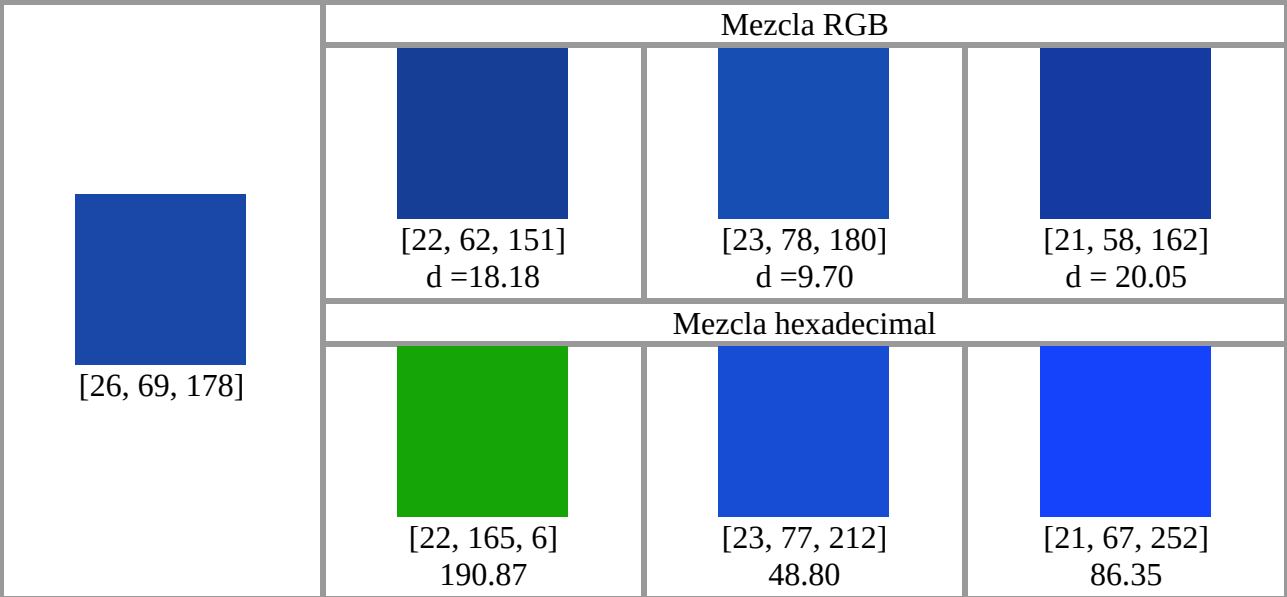
Para poder medir de alguna manera la diferencia con respecto del color original, tratamos a cada color como si fuera un punto en un espacio tridimensional (R,G,B) y hemos calculado la distancia (d) entre los puntos. De acuerdo con esta medida, vemos que los resultados de la primera mezcla parecen más cercanos al original. Sin embargo, salvo algún caso concreto, el resultado de esta mezcla sigue siendo muy distinto al color original.

Original	Normal	Inversa	Aleatoria
71			
			

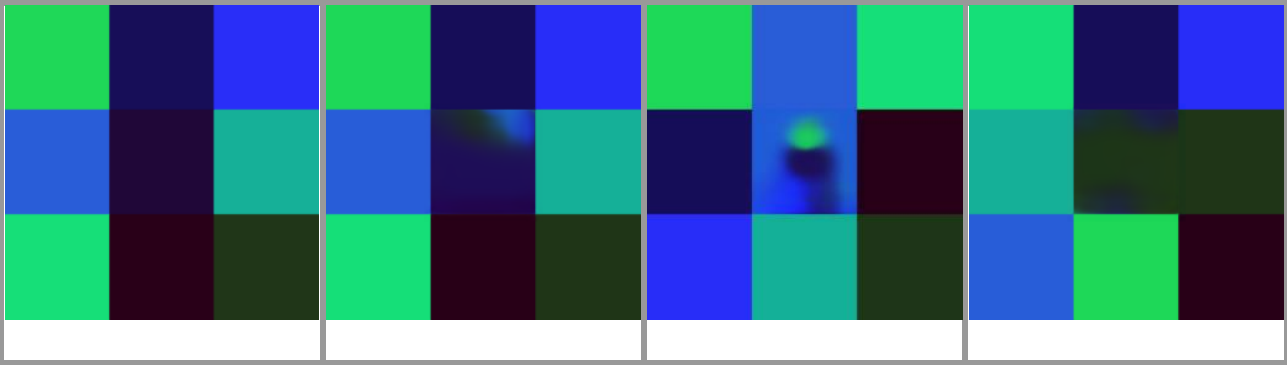
Centros mezclados			
 [0, 55, 86]	Mezcla RGB		
	 [1, 171, 111] d = 118.67	 [1, 172, 101] d = 117.96	 [0, 134, 55] d = 84.86
	Mezcla hexadecimal		
	 [1, 123, 232] d = 150.84	 [1, 52, 10] d = 158.73	 [0, 171, 201] d = 170.17
547			
			
Centros mezclados			
 [8, 100, 164]	Mezcla RGB		
	 [4, 88, 64] d = 100.80	 [4, 80, 57] d = 108.93	 [4, 80, 96] d = 70.99
	Mezcla hexadecimal		
	 [4, 191, 218] 153.87	 [4, 154, 161] 88.79	 [4, 1, 169] 165.97
1657			



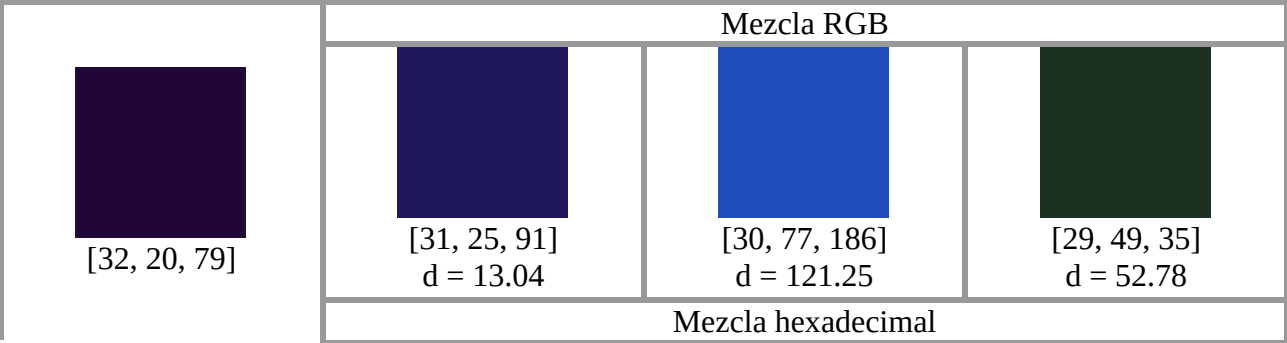
Centros mezclados


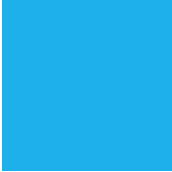



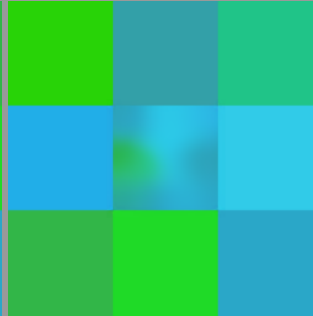
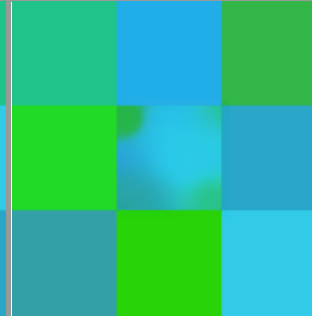


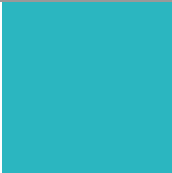
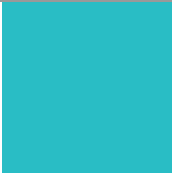

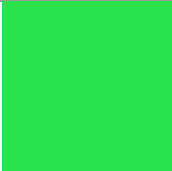


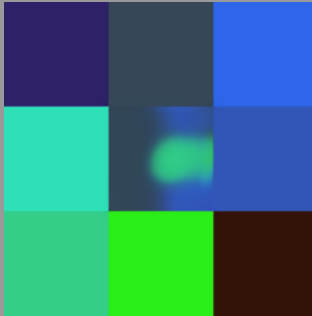









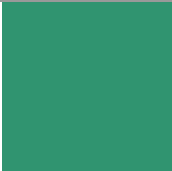



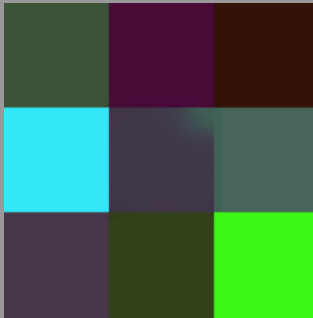
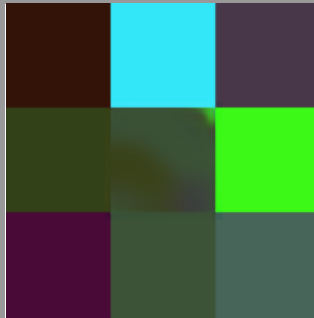
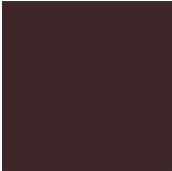





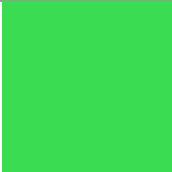
2099

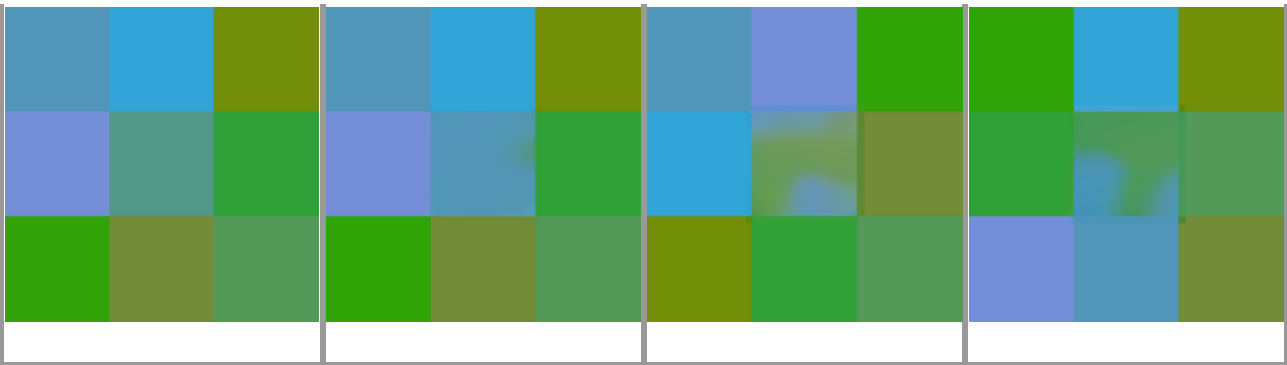


Centros mezclados

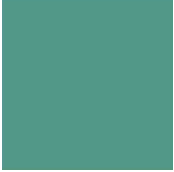

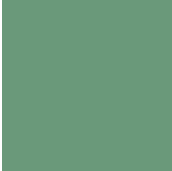
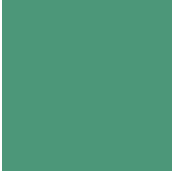

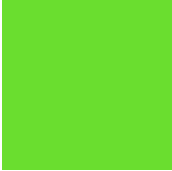



			
	[15, 187, 109] 148.55	[29, 176, 235] 188.27	[29, 53, 96] 17.29
2801			
			
Centros mezclados			
 [43, 184, 119]	Mezcla RGB		
			
	[41, 173, 189] d = 70.89	[43, 182, 192] d = 73.03	[41, 189, 197] d = 78.19
	Mezcla hexadecimal		
			
	[42, 37, 112] d = 146.24	[43, 227, 77] d = 116.11	[41, 89, 147] d = 84.51
3251			
			
Centros mezclados			
	Mezcla RGB		

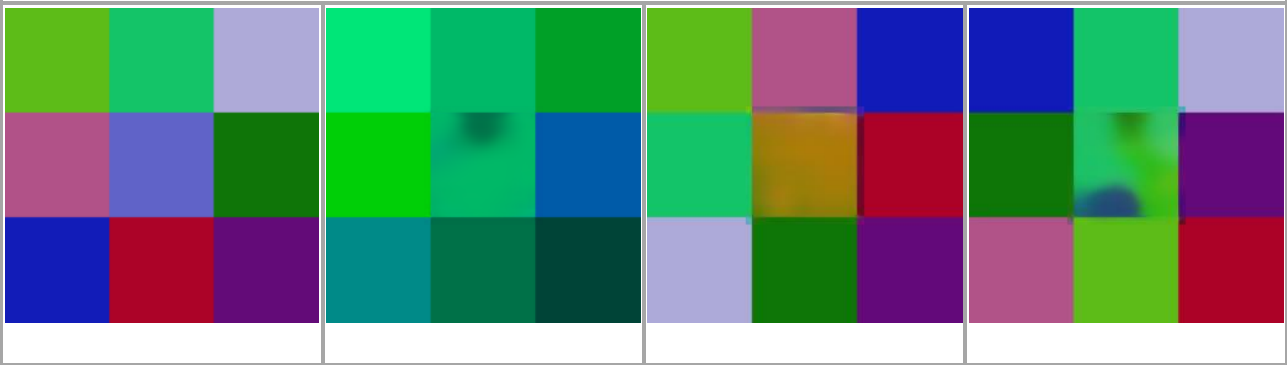
 [49, 159, 75]	 [48, 103, 125] d = 75.08	 [48, 84, 169] d = 120.26	 [49, 33, 104] d = 129.29
	Mezcla hexadecimal		
	 [48, 156, 163] d = 84.56	 [48, 148, 112] d = 48.18	 [49, 79, 1] d = 99.71
4073			
			
Centros mezclados			
 [63, 45, 54]	Mezcla RGB		
	 [61, 73, 74] d = 34.47	 [63, 58, 73] d = 23.02	 [59, 76, 46] d = 32.26
	Mezcla hexadecimal		
	 [61, 139, 131] d = 61.03	 [63, 148, 45] d = 78.01	 [58, 221, 81] d = 132.88
5413			



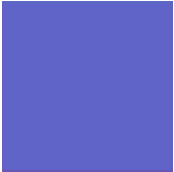



Centros mezclados

 [84, 151, 151]	Mezcla RGB		
	 [82, 150, 180] d = 29.09	 [106, 153, 122] d = 36.46	 [76, 151, 121] d = 31.05
	Mezcla hexadecimal		
	 [82, 202, 5] d = 189.05	 [106, 222, 47] d = 159.45	 [76, 33, 123] d = 134.25

6317



Centros mezclados

 [102, 102, 184]	Mezcla RGB		
	 [96, 130, 20] d = 166.48	 [158, 124, 14] d = 180.33	 [45, 170, 76] d = 139.77
	Mezcla hexadecimal		



[38, 66, 16]
d = 111.76



[46, 78, 173]
d = 99.04



[45, 45, 110]
d = 87.80

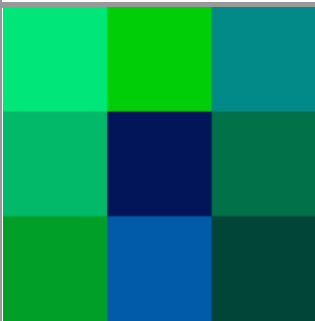
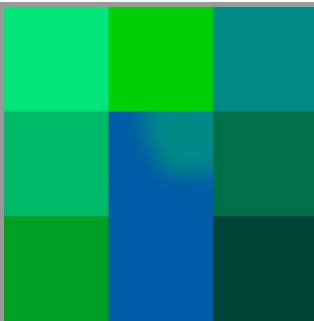
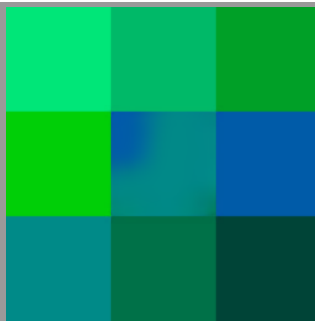
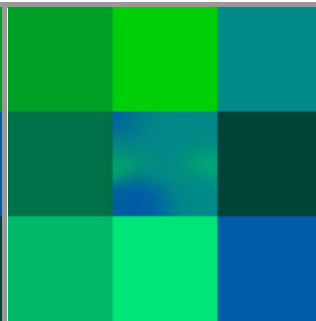


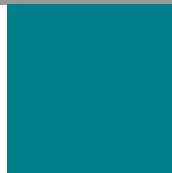
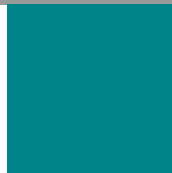

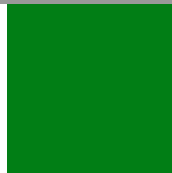
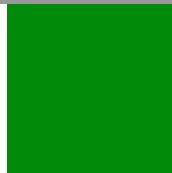

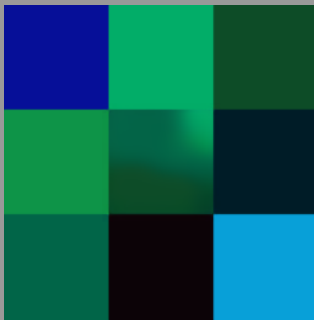
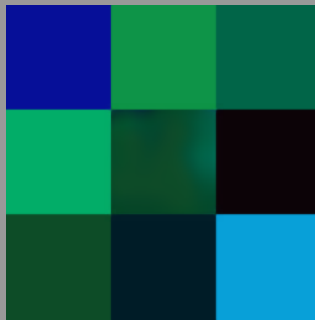
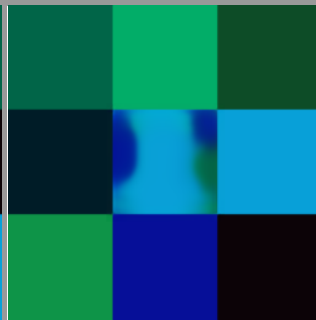
TERCER ENTRENAMIENTO



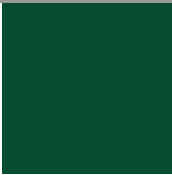


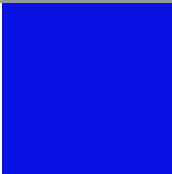

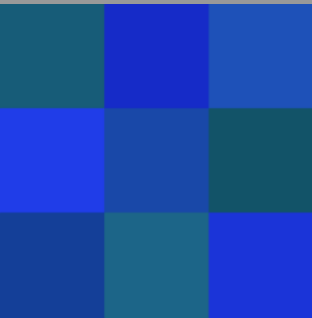
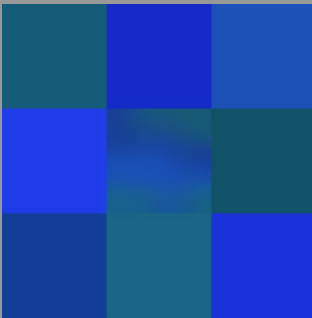
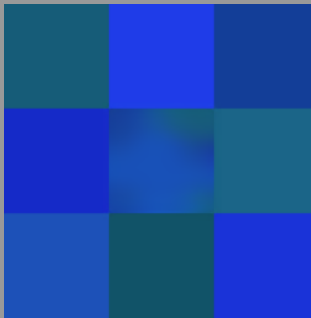
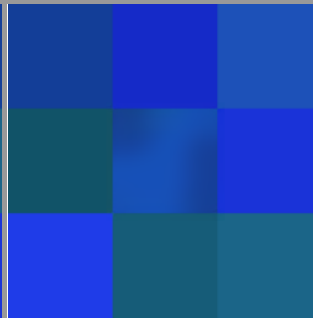





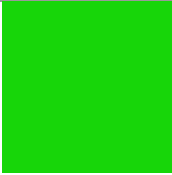
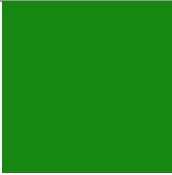
Realizamos un tercer entrenamiento del modelo para matrices que, aunque no fueran de primos, cumplieran la regla de la suma. Esto nos permitió entrenar con un corpus 10 veces más grande.

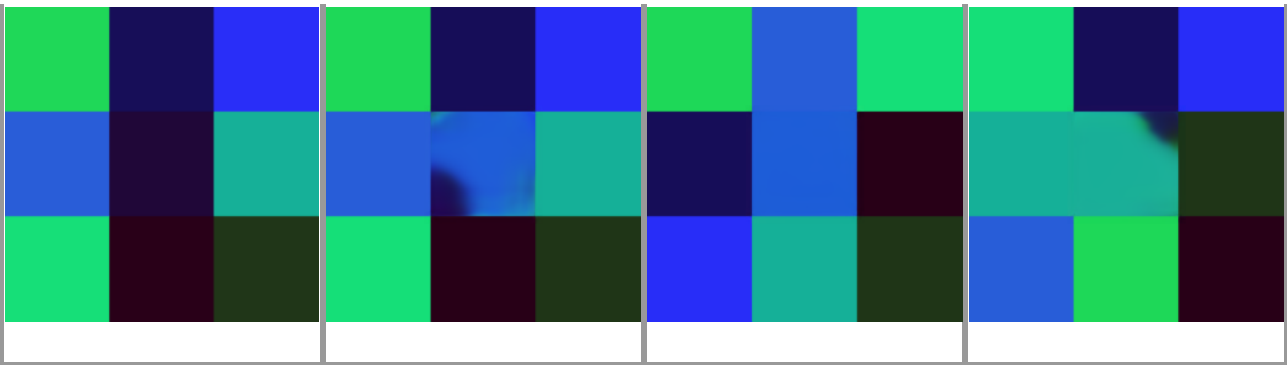
Los resultados mejoraron significativamente con respecto al caso anterior porque, aunque no hiciera hecho una correcta aproximación en la mayoría de los casos, en general los resultados son más cercanos que la última vez. Además, un par de matrices incluso las ha rellenado con un color casi exacto al original.

En el anexo mostramos los resultados del entrenamiento.

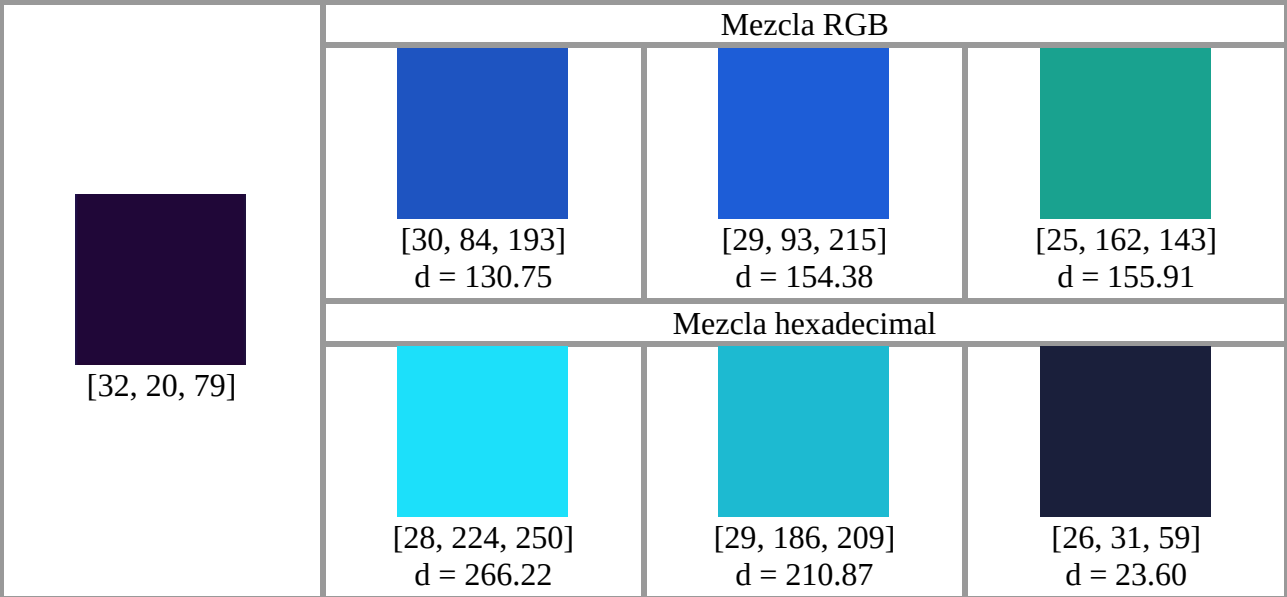
Anexo: Tabla comparación matrices

Original	Normal	Inversa	Aleatoria			
71						
						
Centros mezclados						
 [1, 21, 88]	Mezcla RGB					
						
	RGB	[1, 108, 156]	RGB	[1, 128, 140]	RGB	[1, 131, 138]
	d _{RGB}	110.42	d _{RGB}	118.97	d _{RGB}	120.83
	num	93.34	num	98.444	num	99.21
	d _{num}	22.34	d _{num}	27.444	d _{num}	28.21
	Mezcla hexadecimal					
						
	RGB	[1, 44, 38]	RGB	[1,126,21]	RGB	[1, 137, 9]
	d _{RGB}	55.03	d _{RGB}	124.55	d _{RGB}	140.35
	num	76.838	num	97.813	num	100.617
	d _{num}	5.838	d _{num}	26.813	d _{num}	29.617
547						
						
Centros mezclados						

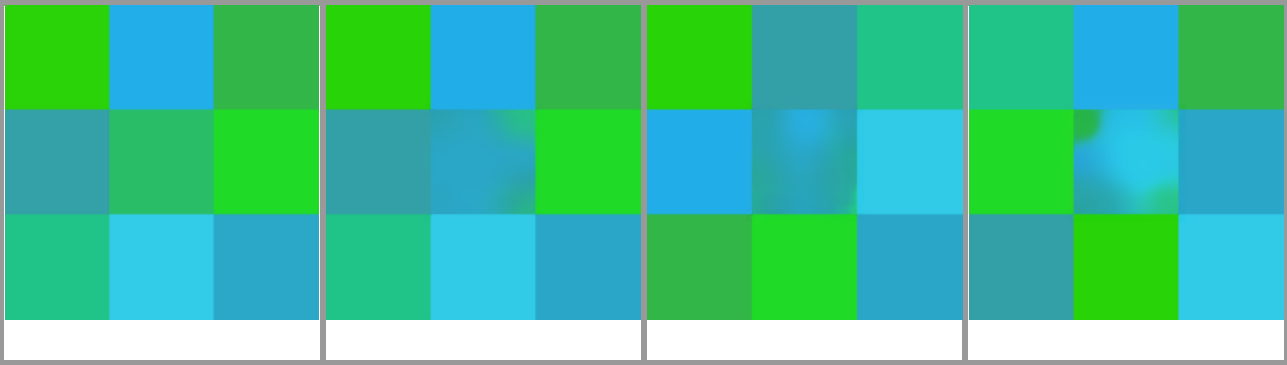
 [8, 100, 164]	Mezcla RGB		
	 [6, 96, 61] d = 103.10	 [8, 77, 47] d = 119.24	 [5, 125, 180] d = 29.83
	Mezcla hexadecimal		
	 [6, 137, 207] d = 56.76	 [8, 18, 226] d = 102.80	 [5, 23, 43] d = 143.45
1657			
			
Centros mezclados			
 [26, 69, 178]	Mezcla RGB		
	 [24, 81, 154] d = 26.91	 [24, 80, 167] d = 15.68	 [22, 75, 171] d = 10.04
	Mezcla hexadecimal		
	 [24, 142, 223] d = 85.78	 [23, 214, 9] d = 222.70	 [22, 135, 17] d = 174.05
2099			



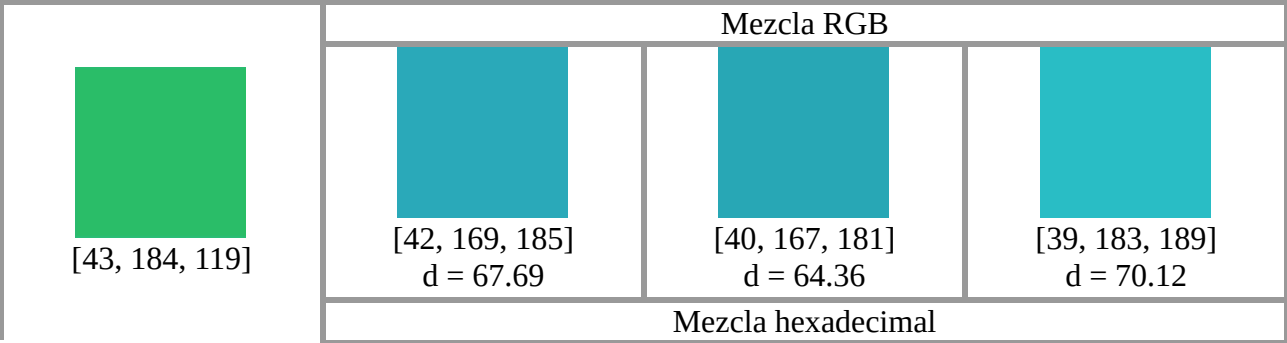
Centros mezclados





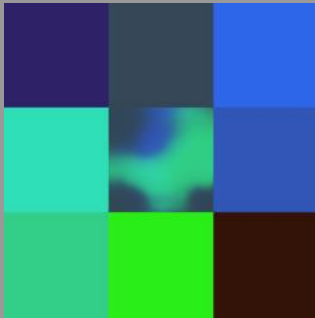
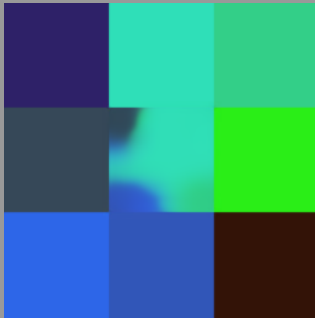
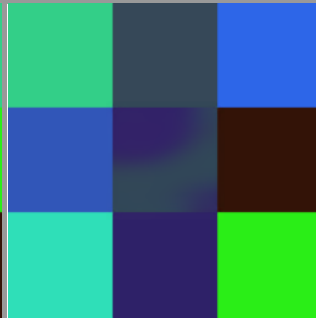
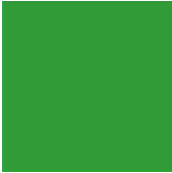
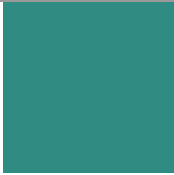
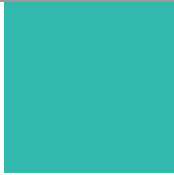





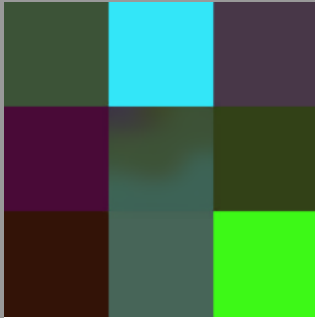
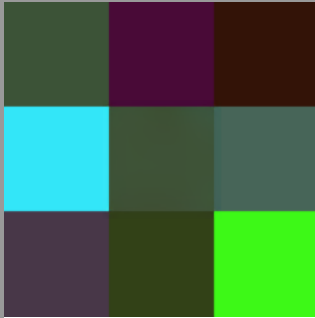
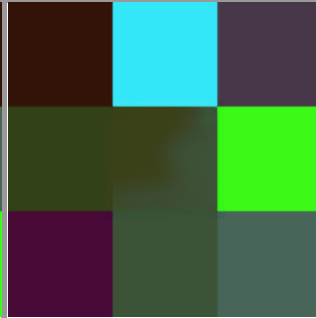


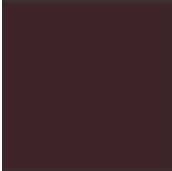

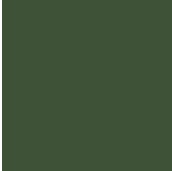


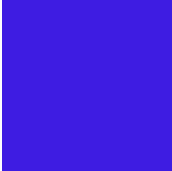

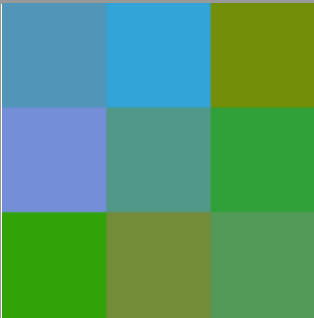
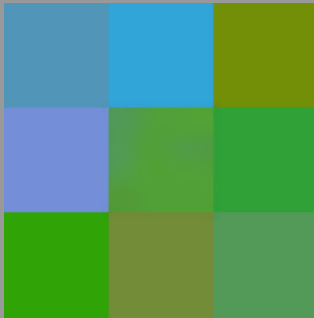
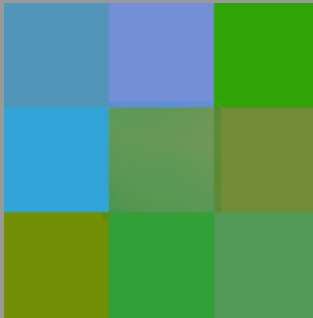

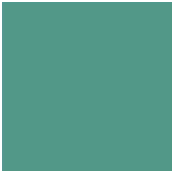






2801

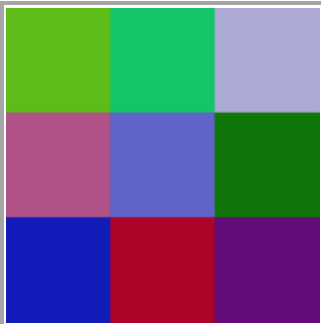
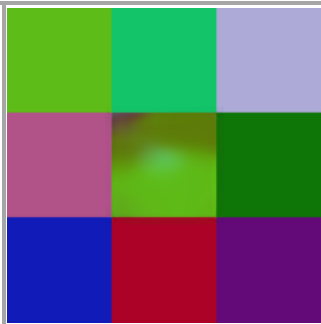
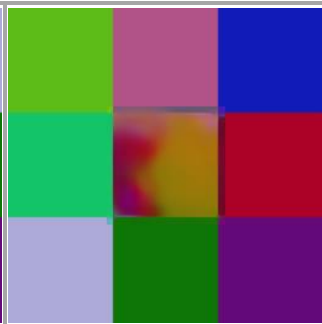
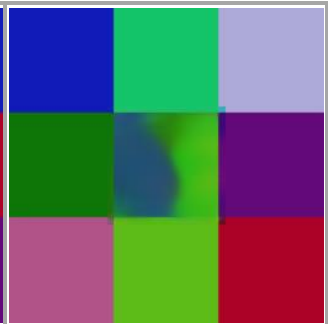
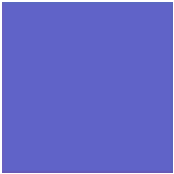



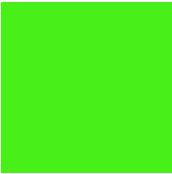
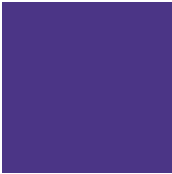
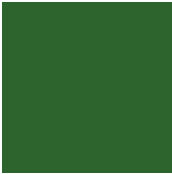


Centros mezclados



			
	[42, 101, 224] d = 133.85	[40, 219, 174] d = 62.26	[39, 87, 219] d = 139.37
3251			
			
Centros mezclados			
 [49, 159, 75]	Mezcla RGB		
			
	[48, 140, 130] d = 58.20	[48, 185, 172] d = 100.43	[49, 55, 95] d = 105.91
Mezcla hexadecimal			
			
	[48, 155, 37] d = 38.22	[48, 207, 137] d = 78.42	[49, 175, 216] d = 141.90
4073			
			
Centros mezclados			
	Mezcla RGB		

 [63, 45, 54]	 [61, 87, 68] d = 44.32	 [62, 82, 55] d = 37.03	 [59, 73, 40] d = 31.56
	Mezcla hexadecimal		
	 [61, 148, 162] d = 149.25	 [62, 28, 226] d = 172.84	 [59, 114, 35] d = 71.68
5413			
			
Centros mezclados			
 [84, 151, 151]	Mezcla RGB		
	 [81, 158, 62] d = 89.33	 [96, 152, 86] d = 66.11	 [75, 151, 187] d = 37.11
	Mezcla hexadecimal		
	 [81, 150, 238] d = 87.06	 [96, 196, 76] d = 88.28	 [75, 127, 111] d = 47.51
6317			

			
Centros mezclados			
 [102, 102, 184]	Mezcla RGB		
	 [95, 157, 25] d = 168.39	 [150, 84, 37] d = 155.68	 [46, 127, 67] d = 132.10
	Mezcla hexadecimal		
	 [73, 239, 25] d = 211.87	 [75, 53, 134] d = 75.03	 [45, 100, 45] d = 150.25