|  |  |  |
| --- | --- | --- |
| Algoritmo | Parametros | Cantidad de Clusters |
| DBScan | eps = 0.5 min\_samples= 5 | 0 |
| eps = 0.5 min\_samples =2 | 3 |
| eps = 150, min\_samples = 4 | 19 |
| MeanShift | Default | 12 |

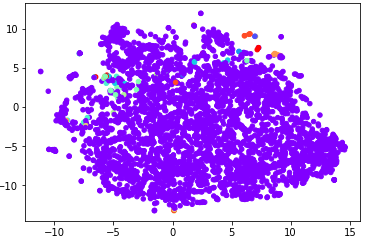
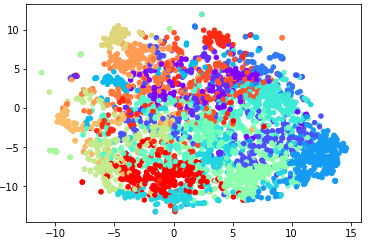
Se comienza utilizando PCA con el fin de disminuir la cantidad de dimensiones de cada imagen, sin alterar de manera significativa el estudio de estas. El algoritmo deja afuera las dimensiones con menor importancia (en términos de varianza) y nos deja para estudiar las 50 principales dimensiones, logrando que los cálculos sean bastante más fáciles de realizar.

|  |  |  |
| --- | --- | --- |
| Algoritmo | Parametros | Silhouette score |
| Kmeans | K=10 | 0.0540114168266 |
| K=20 | 0.0540319770445 |
| K=30 | 0.0550646632473 |
| DBScan | eps = 0.5 min\_samples= 5 | Indefinido |
| eps = 0.5 min\_samples =2 | -0.100149531082 |
| eps = 150, min\_samples = 4 | -0.194101650753 |
| MeanShift | Default | 0.215734653618 |

El valor del silhouette score para DBScan con los parametros por default no se puede calcular ya que se necesitan al menos dos clusters.

Si bien se observa en la tabla que MeanShift tiene un valor bastante más alto de Silhouette score que Kmeans, al analizar los gráficos de ambos, llegamos a la conclusión de que Kmeans, con K=20 es el mejor algoritmo para generar los clusters. Creemos que esto se debe a que MeanShift determina por si mismo la cantidad de clusters, encontrando que la manera mas consistente de agrupar los datos es como se ve en la imagen, es posible que esto suceda porque este algoritmo busca “blobs” y es probable que los datos no estan distribuidos de esta manera. La diferencia que tiene Kmeans es que al definir la cantidad de clusters que buscamos, este algoritmo es forzado a encontrar más agrupaciones de datos que probablemente sean menos consistentes con una distribucion de “blobs”, de hecho asi lo muestra el silhouette score, pero que al observar los resultados en la visualizacion nos damos cuenta que es mejor para clasificar las imágenes.

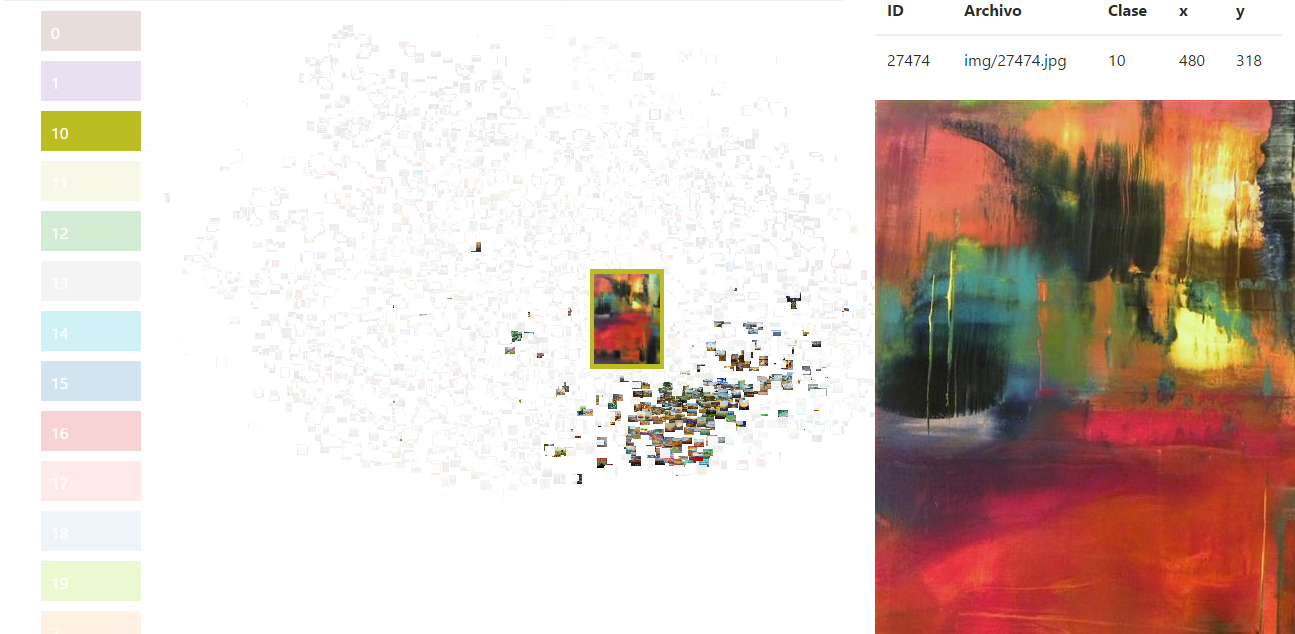
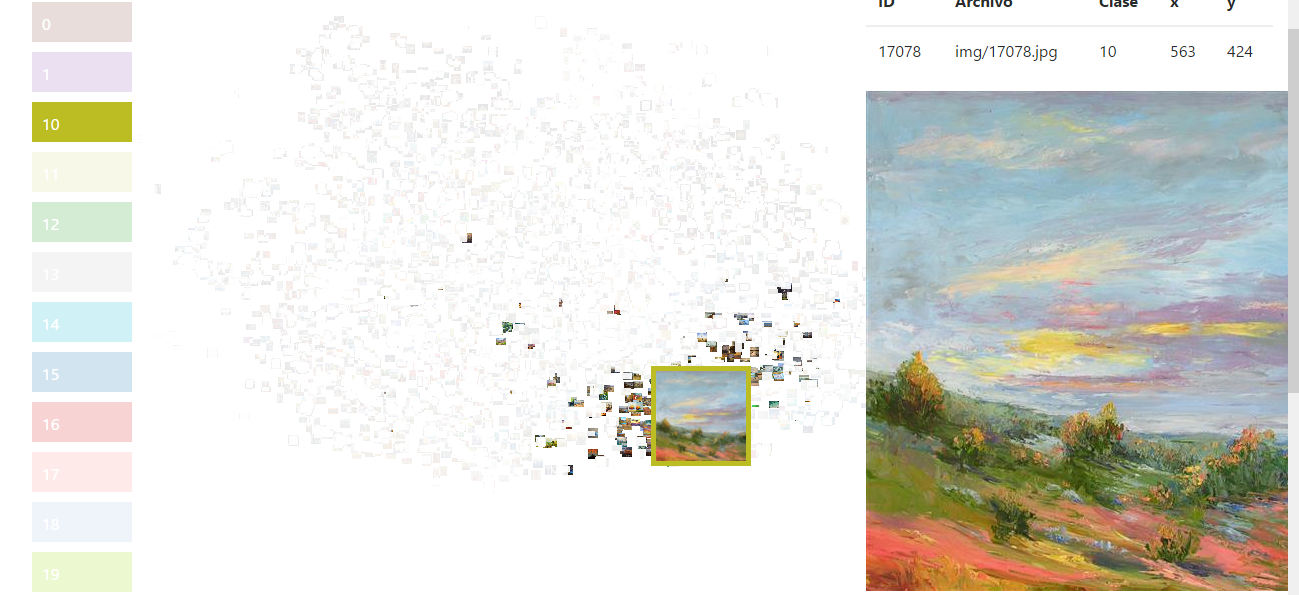
Gráficos de Kmeans con K=20 y MeanShift respectivamente:



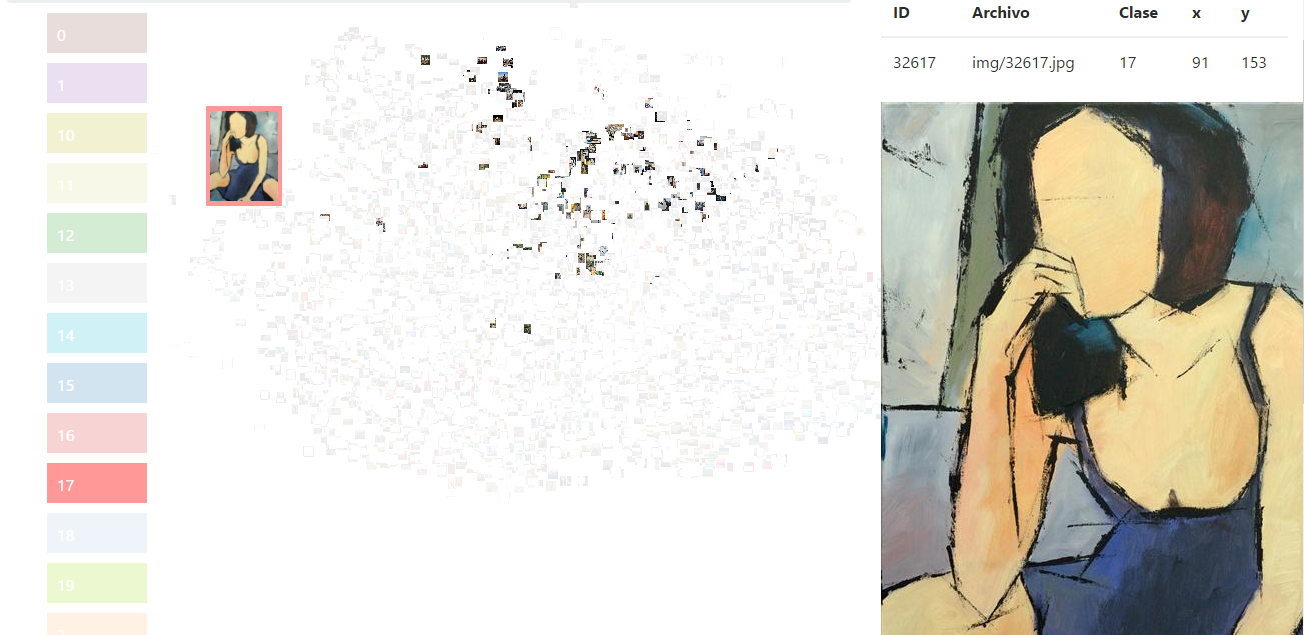
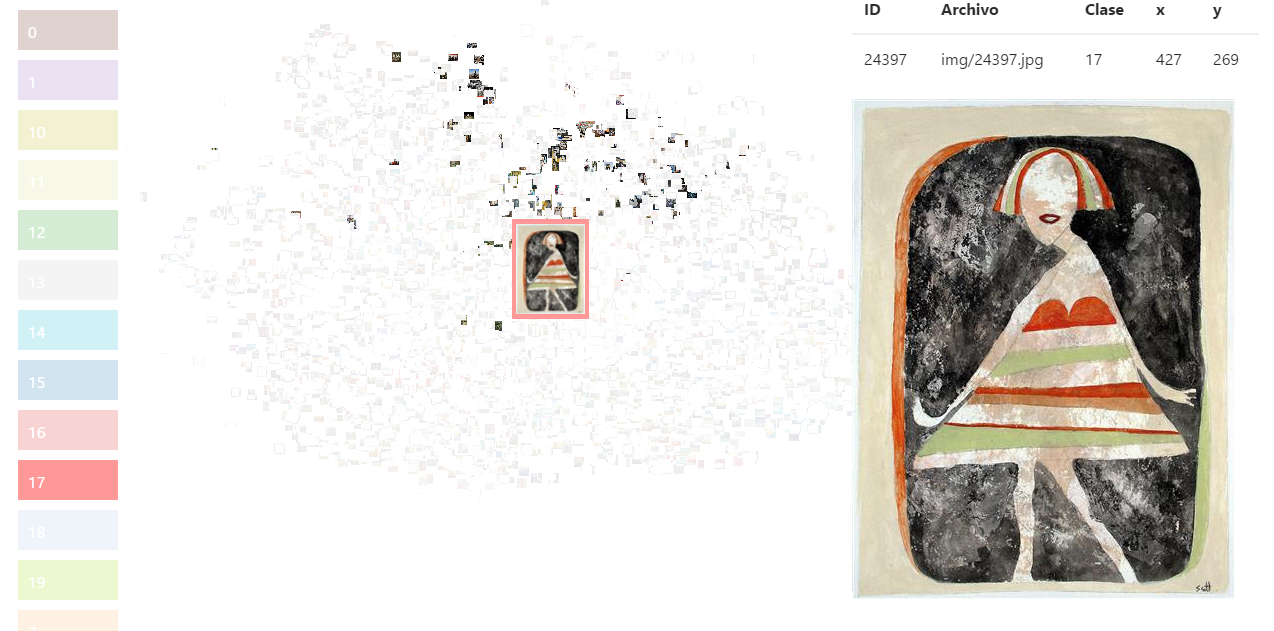
Como se observa, Meanshift genera un cluster que contiene a la mayoría de las imágenes y 11 clusters con una sola imagen. Por esta misma razón tiene un silhouette score tan alto, pero evidentemente no es lo que buscamos. Por otro lado, Kmeans con K=20 nos muestra algo bastante más razonable en términos de clustering.

Los clusters seleccionados toman en consideración distintas características de las imágenes, tales como:

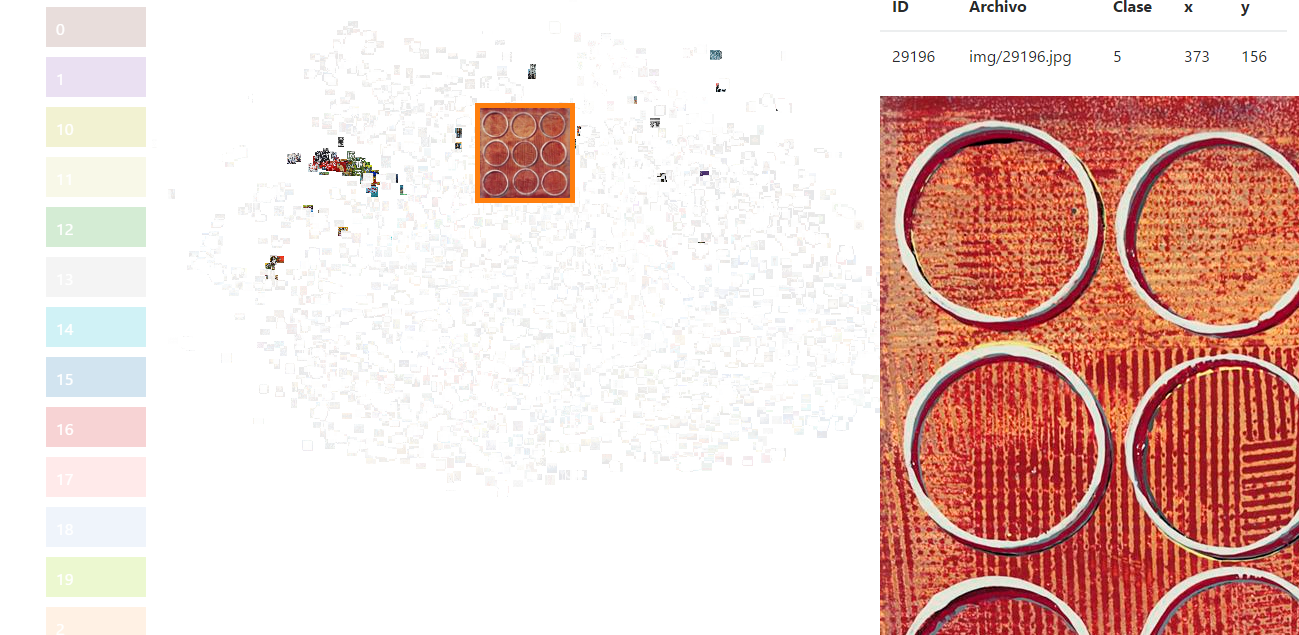
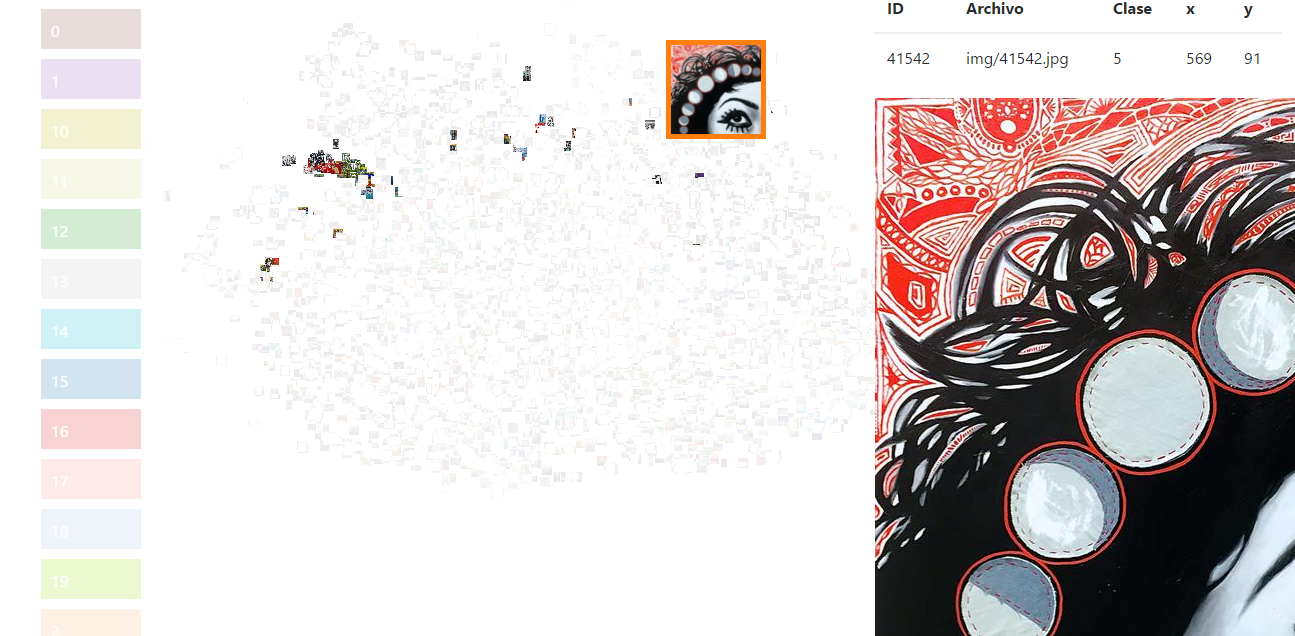
- El estilo de pintura:



* La presencia de ciertas figuras (como mujeres o siluetas):



* Repetición de formas:



Se observa que los clusters se generaron a la mezcla de características como estas, sumadas a otras como colores, tipos de formas y diferenciación entre fotos y pinturas. De esta manera, ciertas imágenes que comparten algunas de estas características son clasificadas en los mismos clusters.