

Reconocimiento de personajes en League of Legends

Grupo 1.1
Isidro Alfaro Barbará
Ismael Barbeito Vázquez
Rodrigo Martín Prieto
Daniel Ruíz Perez

1 Introducción

El mundo de los eSports (deportes electrónicos, en su acrónimo inglés) está cobrando cada vez un mayor auge en nuestros días. Decenas de equipos, formados por cientos de jugadores, patrocinadores y empresas de publicidad tratan de conseguir un trozo del pastel que supone esta nueva forma de negocio. Los partidos de videojuegos como Counter Strike o StarCraft2 son retransmitidos en todo el mundo, y las grandes finales mundiales empiezan a coger la fama de sus equivalentes en lo que algunos ya llaman deporte físico o tradicional.

En este marco, es notable el increíble crecimiento del videojuego League of Legends (o LoL, como los aficionados lo abrevian), creado por la empresa Riot Games. El LoL pertenece al género de los MOBA (Multiplayer Online Battle Arena), donde dos equipos de cinco jugadores se batén en un campo restringido para intentar destruir el edificio clave de la base enemiga, conocido como Nexus. Bajo una fórmula aparentemente tan sencilla se esconde un juego rico en detalles a nivel técnico y estratégico, con 120 personajes únicos, y muy complejo de dominar.



A fecha de octubre de 2012, Riot anunció que el LoL había batido la cifra de 70 millones de jugadores registrados entre todos sus servidores, repartidos a lo largo del mundo. La tercera edición del torneo mundial del juego entre los mejores equipos del mundo, celebrada en 2013, contó con 32 millones¹ de espectadores a través de los diversos streamings realizados en multitud de idiomas.

Con todo esto en mente, el equipo de trabajo consideró interesante usar el juego como base para el presente proyecto. En concreto, se investigará en la identificación de los diferentes personajes de League of Legends en los mapeados del mismo, utilizando para ello redes de neuronas artificiales, máquinas de soporte vectorial y k-NN. Un buen sistema de reconocimiento de personajes a partir de imágenes tomadas del juego (*ingame*) podría ser la base para diversos mecanismos que ayuden a mejorar la calidad de la información que se puede proporcionar en directo en las retransmisiones (por ejemplo, ser capaz de reconocer la velocidad a la que los jugadores controlan a sus personajes, reconocimiento de patrones de ataque en peleas de equipo entre la multitud de efectos visuales generados...).

2 Descripción del problema

Se plantea, pues, un sistema capaz de reconocer personajes "in-game". Hay que tener en cuenta el gran número de personajes diferentes, más de cien, y que cada personaje dispone de 2 hasta incluso 7 u 8 aspectos alternativos. Como facilidades añadidas, se dispone de una base de datos que permite ver los modelos en 3D de todos los personajes y aspectos, y que el juego toma siempre el mismo ángulo de visión (isométrica) lo cual facilita crear una base de datos consistente para el entrenamiento del sistema.

Para delimitar el dominio del trabajo, resulta conveniente establecer una serie de restricciones, como limitar drásticamente el elenco de personajes a reconocer y el tipo de fondos en la medida de lo posible.

Para la construcción de la base de datos se tomarán capturas del juego. Se han separado en dos tipos: aquellas que representan casos positivos (es decir, personajes) y aquellas que representan casos negativos (elementos del decorado, fondo, la interfaz de usuario o HUD...). Ambas están compuestas por recortes de las imágenes obtenidas del juego. Juntando capturas de cada tipo se conforma el conjunto de entrenamiento.

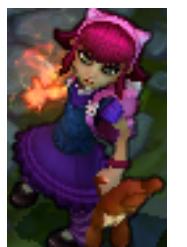


Fig 2.a y 2.b: recortes de ejemplo (personaje y fondo)

Se tendrán ficheros que seguirán la siguiente estructura:

<NombrePJ>_<codFondo>[hud]<numAprox>_<numImgs>.mat

Como convenio, se establece que los códigos para cada fondo serán:

std - Fondo para líneas: Tierra, verde, plantas etc.

riv - Fondo para el río y zonas acuosas.

bas - Fondo para la base, edificaciones y zonas adoquinadas.

Se usarán nombres en mayúscula, se eliminarán caracteres especiales como apóstrofes.

Por ejemplo, BLITZ_STD_7_50.mat indica que la base de datos contiene 50 imágenes con fondo estandar del personaje llamado BLITZ, extrayendo las características indicadas en la

aproximación 7.

De cada imagen, se extraerán características relacionadas con los canales RGB de la misma. Esto es válido para este proyecto debido al alto contraste de paletas utilizadas entre los fondos de los mapas y los personajes siendo los primeros más apagados (grises, marrones, azules oscuros y verdes), mientras que los personajes utilizan tonos más vivos. Sería interesante identificar otros valores tales como siluetas, patrones o elementos dentro de las imágenes debido a la gran variedad de entrada.

3 Bibliografía/Estado del Arte

No hay antecedentes conocidos para el League of Legends en este campo. Es por ello que, como ya se ha mencionado, el equipo de trabajo ha creado los recursos a usar (capturas de dentro del propio juego) al no poder encontrar ninguna base de datos útil ya creada para este problema.

Lo más similar que se ha encontrado es un trabajo en reconocimiento de imágenes pensado como base para un sistema de inteligencia artificial capaz de jugar al Starcraft II (Para el videojuego Starcraft II), creación de Matthew Fisher². Se utiliza, entre otras técnicas, procesado de texturas para intentar recrear la situación actual del juego, y que luego un sistema inteligente proporcione una respuesta acorde con la situación del juego. La idea básica es que el sistema captura las llamadas al renderizado de texturas que realiza el propio juego para mantener una base de conocimientos actualizado de las mismas. Así mismo, se mantiene una relación de las unidades reconocidas y su estado, el minimapa... Con todo esto, un sistema de IA realiza la toma de decisiones.

Aunque este trabajo podría ser una base interesante para aproximaciones más avanzadas, se ha considerado excesivamente complejo para el dominio de la asignatura.

4 Desarrollo

Se ha planteado el desarrollo del problema en aproximaciones. Cada aproximación varía en las características tomadas de las imágenes. Actualmente, se han llevado a cabo 5 aproximaciones.

Resulta interesante hacer mención en que, aunque las bases de datos están compuestas por recortes de capturas de pantalla, el sistema recibe siempre imágenes completas. Se necesita que, al procesarlas para identificar si hay personaje o no (y donde), se hagan “ventanas”. Una ventana es un rectángulo que representa una sección de la imagen original. El programa genera muchas ventanas de diferentes dimensiones que recorren toda la captura de forma secuencial. Posteriormente son enviadas al clasificador escogido para que determine si pertenecen o no a un personaje. En caso de que reconozca un positivo, recuadra dicha ventana en el original.

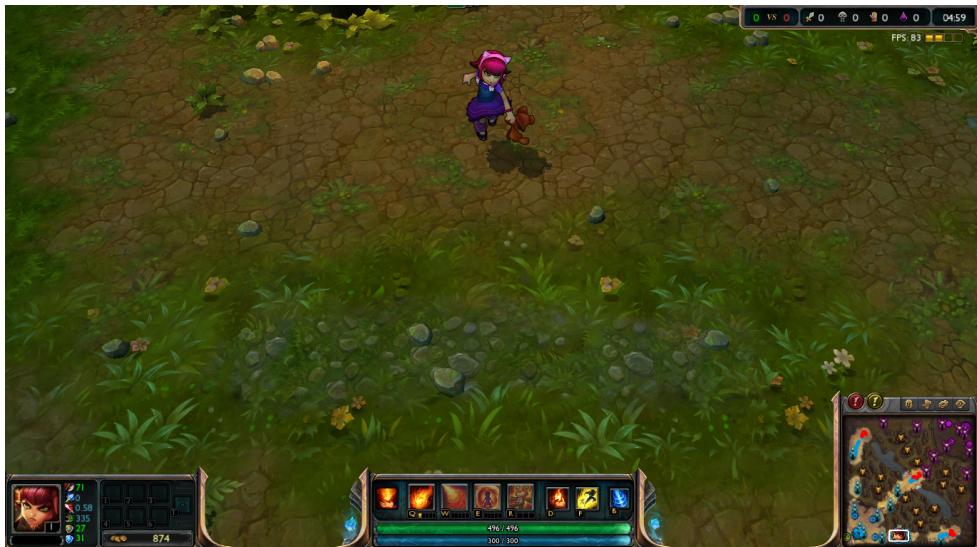


Fig 4.a: ejemplo de imagen recibida por el sistema

Las ventanas se amplían de forma independiente en la X y en la Y, pero siempre en una cantidad fija, que se ha denominado desplazamiento. Se ha considerado el uso de tres desplazamientos: 10, 25 y 40.

Como se comentó en la introducción, se usan tres tipos de clasificadores distintos: redes de neuronas, SVM con varios kernels (Linear, MLP, Polynomial, Quadratic, RBF) y KNN para varios valores de k(1, 5 y 9). Cada aproximación ha sido testeada bajo cada variante de los clasificadores.

Cabe mencionar que en el caso de RNA, se obtiene, salvo algún error (que se vuelve a ejecutar), un 100% de entrenamiento, y en SVM un 100% de clasificación. Esto es debido al

enorme contraste de colores entre el fondo y los personajes, como ya se ha comentado. Debido a esto, no se mencionará este aspecto a lo largo de la memoria.

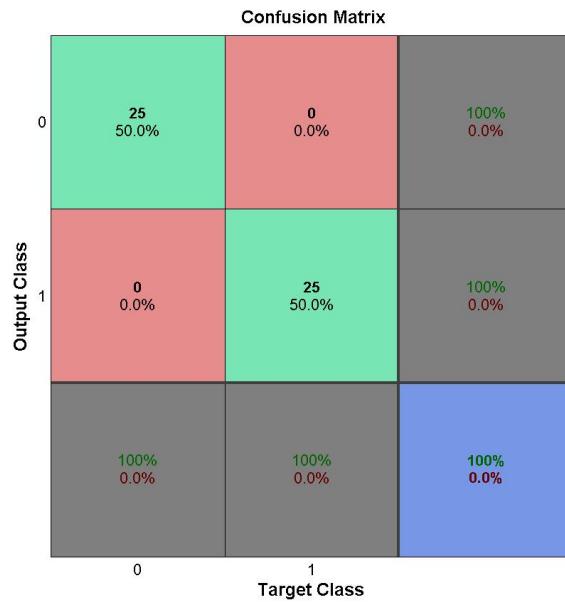


Fig 4.b: muestra de matriz de confusión para una red bien entrenada

A continuación se presenta una relación de las diferentes aproximaciones consideradas a lo largo del proyecto.

Aproximación 1

1.1.-Contenido

La aproximación 1 contempla únicamente la media de los canales RGB de cada imagen. Esto es, cada base de datos es una matriz de $3 \times n$ elementos, siendo n el número de imágenes.

1.2.- Recursos usados

A la hora de realizar la aproximación, han sido usadas las bases de datos ANNIE_STD_1_25.mat para los casos positivos y FONDO_STDHUD_1_25.mat para los casos negativos. En un principio se utilizó FONDO_STD_1_25.mat, pero al no contemplar el HUD se producían muchos falsos positivos en el minimapa, la interfaz de usuario...

1.3.- Ejecuciones

A) Redes de neuronas

Las redes de neuronas se comportan bastante bien con esta aproximación. Se logra un encuadrado muy bueno con desplazamiento 10, a costa de un enorme tiempo de ejecución (sobre una hora y media). Con los otros desplazamientos el resultado es quizás menos preciso (en el ejemplo se pierde una parte de la zona inferior), pero en tiempo es muchísimo mejor.



Fig 4.1.1 y 4.1.2: RNA con desplazamientos 10 y 40, respectivamente

B) SVM

SVM presenta resultados muy dispares. Cuenta con los mejores y los peores encuadrados obtenidos, en función del kernel. El que peor funciona parece ser linear, mientras que los dos mejores son MLP y RBF, siendo el primero quizá un poco mejor. A nivel genérico parece funcionar mejor con desplazamientos altos. Sin embargo, presenta varios falsos positivos (más acuciados con desplazamientos bajos), cosa que en RNA no aparecía. Sus tiempos son los más rápidos de entre todos los clasificadores.

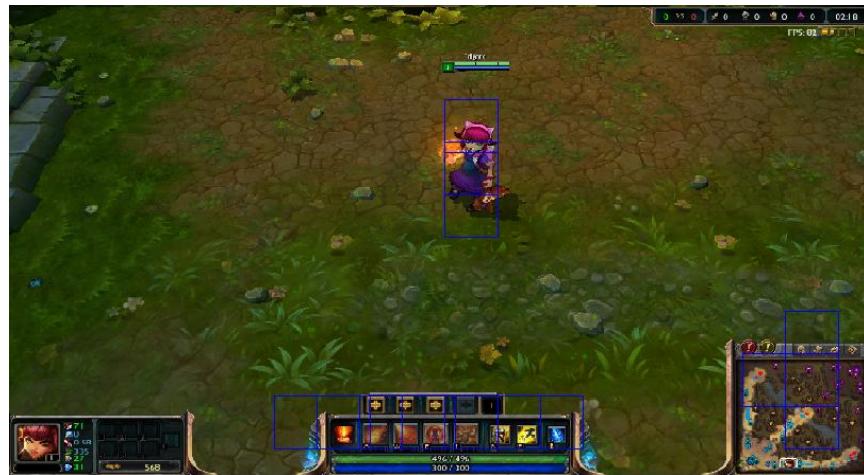


Fig. 4.1.3: Kernel Polynomial con desplazamiento 40

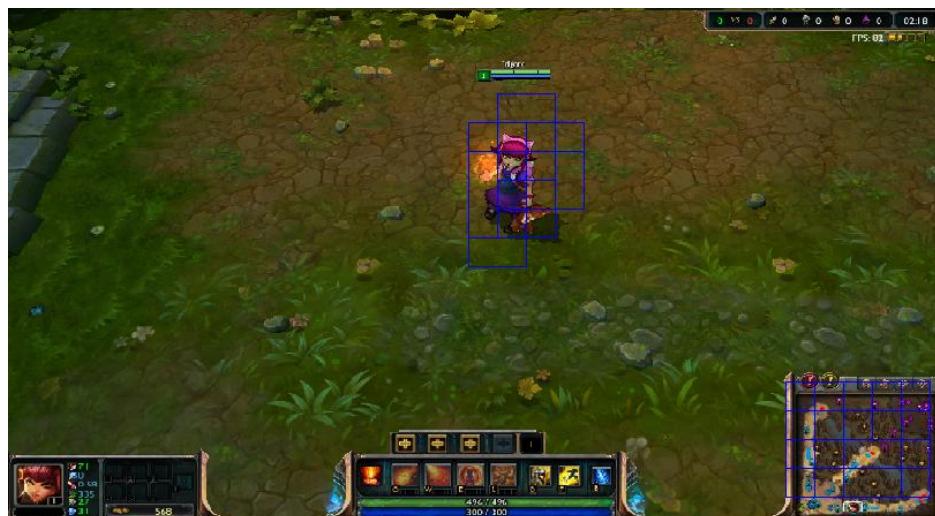


Fig 4.1.4: Kernel MLP con desplazamiento 25

C) KNN

KNN es la peor de las tres técnicas en esta aproximación. De los tres valores de k tomados, el que mejor parece funcionar en todos los casos es k = 1. Entre k = 5 y k = 9 no hay demasiada diferencia, aunque el primero parece reconocer un poco mejor al personaje. Además, tarda tres veces más que SVM.

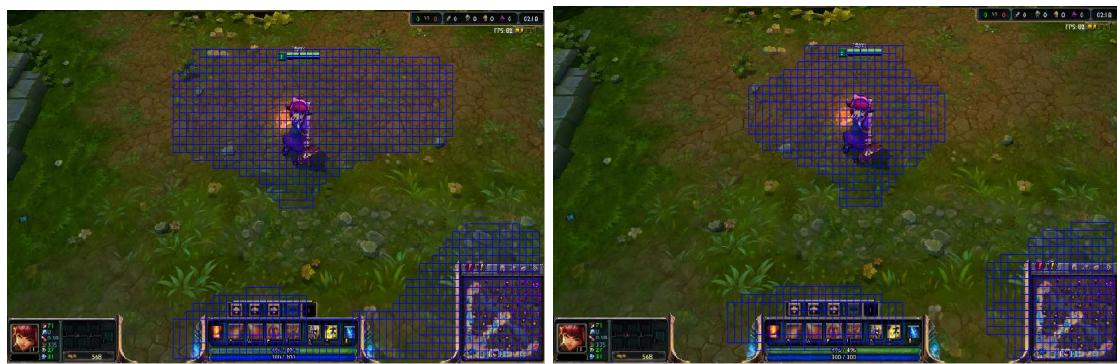


Fig: 4.1.5 y 4.1.6: Resultados para $k = 9$ y $k = 1$ con desplazamiento 10.

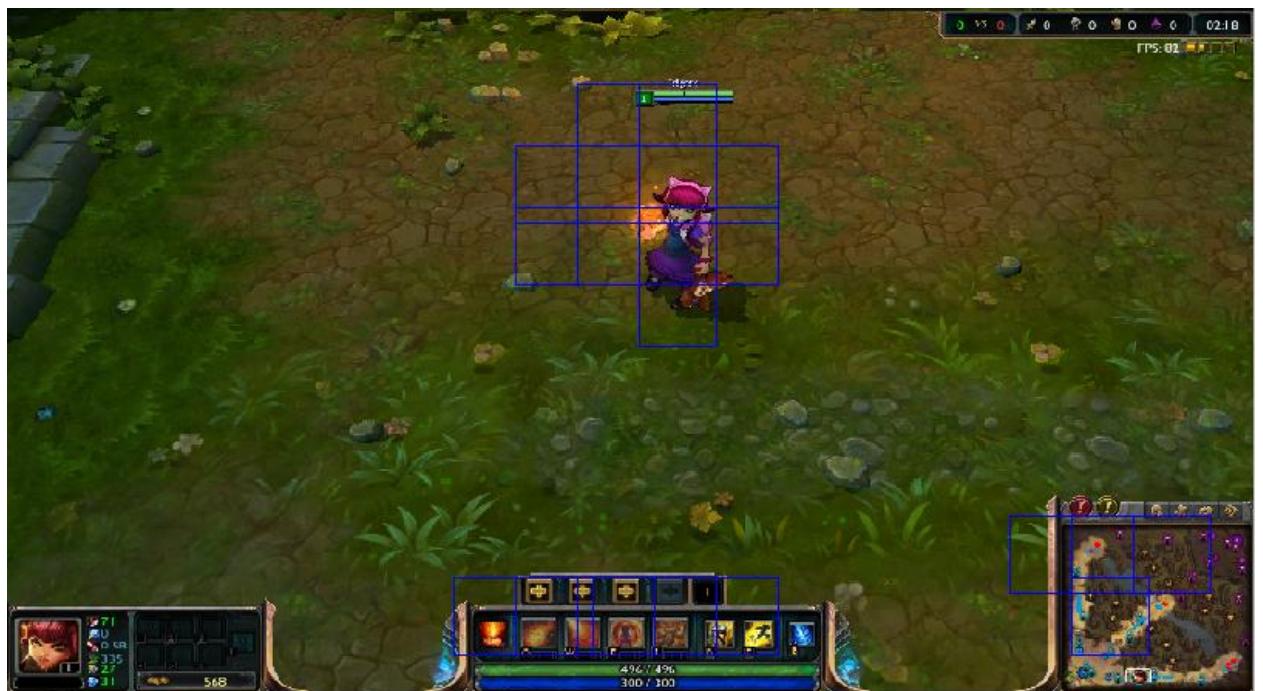


Fig 4.1.7: Mejor ejecución obtenida con KNN ($k = 1$ con desplazamiento 40)

1.4.- Conclusiones

La aproximación ofrece un resultado aceptable para ser la primera iteración del trabajo. Sin embargo, al estar considerando sólo las medias, en cuanto un fragmento de Annie aparece recuadrado en alguna ventana considera que esa zona es un positivo.

Para siguientes aproximaciones se intentará seguir en la misma línea de trabajo, quizá buscando un incremento de características usadas, para tratar de discriminar mejor.

Aproximación 2

2.1.-Contenido

La aproximación 2 es una mejora inmediata sobre la 1. Se calculan, además de las medias RGB, la desviación típica para cada canal. Cada imagen se representa, pues, con seis características, por lo cual se generan matrices de $6 \times n$ elementos, siendo n el número de capturas.

2.2.- Recursos usados

A la hora de realizar la aproximación, han sido usadas las bases de datos ANNIE_STD_2_25.mat para los casos positivos y FONDO_STDHUD_2_25.mat para los casos negativos.

2.3.. Ejecuciones

A) Redes de neuronas

En relación a la aproximación anterior, se percibe un cierto empeoramiento en desplazamientos bajos. Sin embargo, los resultados con desplazamientos altos parecen ser mejores que en el caso de la aproximación 1. En general, las neuronas presentan mayor éxito con esta aproximación que con la anterior.



Fig 4.2.1 y 4.2.2 : RNA para desplazamientos 40 y 10, respectivamente

B) SVM

RBF, Quadratic y Linear dan excelentes resultados. En general, el número de falsos positivos ha disminuido bastante. Se obtienen recuadrados bastante precisos, y en algunos casos se llega a tener un número muy pequeño de ventanas, con una precisión más que aceptable. La mejora respecto a la aproximación 1 es notable.

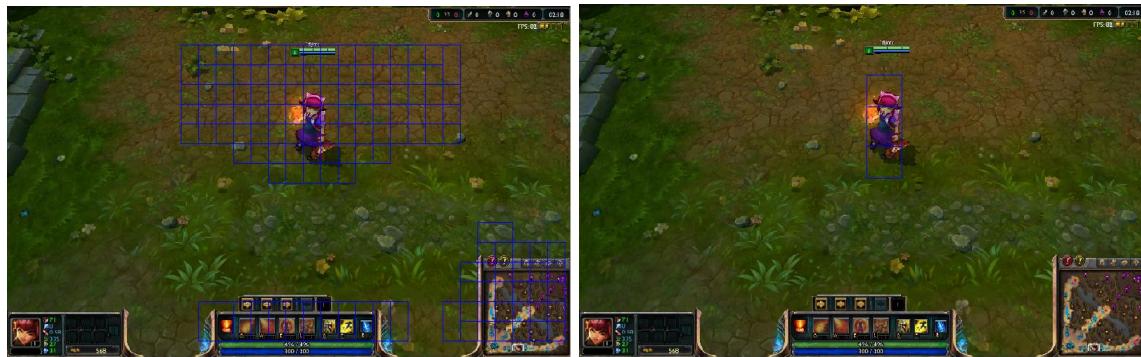


Fig. 4.2.3 y 4.2.4: Polynomical desp 25 y MLP desp 40

C) KNN

KNN se mantiene de nuevo como el peor clasificador. Aún así, en esta aproximación se logra reducir el número de falsos positivos y mejorar un poco la precisión al recuadrar. De todos modos, sigue identificando áreas demasiado extensas, y con $k = 5$ y 9 sigue dando problemas con el minimapa.



Fig. 4.2.5 y 4.2.6: Desplazamientos 40 y 10 para $k = 1$

2.4.- Conclusiones

Simplemente añadiendo las desviaciones típicas, la mejora es importante. En general, se percibe un ajuste mejor del personaje y una reducción del número de falsos positivos. Se puede decir que es una aproximación prometedora.

Llegados a este punto, se presentan dos posibles ramas de trabajo: intentar mejorar el reconocimiento para un personaje concreto o probar la técnica para más personajes y fondos. Las siguientes tres aproximaciones trabajan en la línea de tratar de ajustar a la niña usada en las dos anteriores (Annie).

Se define un rasgo como un aspecto físico relevante del mismo. En Annie rasgos podrían ser el pelo, la bola de fuego, el oso de peluche... Cada uno de ellos está determinado por seis características de color (3 medias y 3 desviaciones), relativas a la sección de la captura en la que aparece. Por ejemplo, si se quisiera tomar el pelo y el oso, se necesitarían 12 características.

Aproximación 3

3.1.-Contenido

La aproximación 3 presenta doce características. Las seis primeras son las mismas que las que usaba la aproximación 2. Las otras seis son también medias RGB y desviaciones, pero para el rasgo que representa el pelo de Annie, detectado en el tercio superior de la imagen.

3.2.- Recursos usados

A la hora de realizar la aproximación, han sido usadas las bases de datos ANNIE_STD_3_25.mat para los casos positivos y FONDO_STDHUD_3_25.mat para los casos negativos.

3.3.- Ejecuciones

A) Neuronas

Empeora de forma drástica respecto a la aproximación anterior. Es incluso peor que en la primera con desp = 10, donde causa falsos positivos, algo que no había pasado antes. Ni con desplazamientos grandes es incapaz de detectar de forma correcta a Annie.



Fig 4.3.1 y 4.3.2: RNA para desplazamientos 40 y 10, respectivamente

B) SVM

Si bien no es tan rotundo como en el caso de las neuronas, en este clasificador también se percibe un empeoramiento en los resultados, aunque una vez más depende del kernel. Quadratic, por ejemplo, mejora bastante, pero Linear, RBF y MLP empeoran mucho. De todos modos, los mejores resultados con esta aproximación siguen sin superar a los obtenidos con la aproximación 2.



Fig. 4.3.3 y 4.3.4: Polynomial desp 25 y Quadratic desp 10

C) KNN

En KNN la aproximación 3 aporta mejores resultados que en la 2. Los falsos positivos pasan a quedar localizados solo en el minimapa, mientras que los ajustes de Annie se mantienen igual o mejores que en el caso de la aproximación 2. Es quizás la única mejora sustancial que aporta esta aproximación.

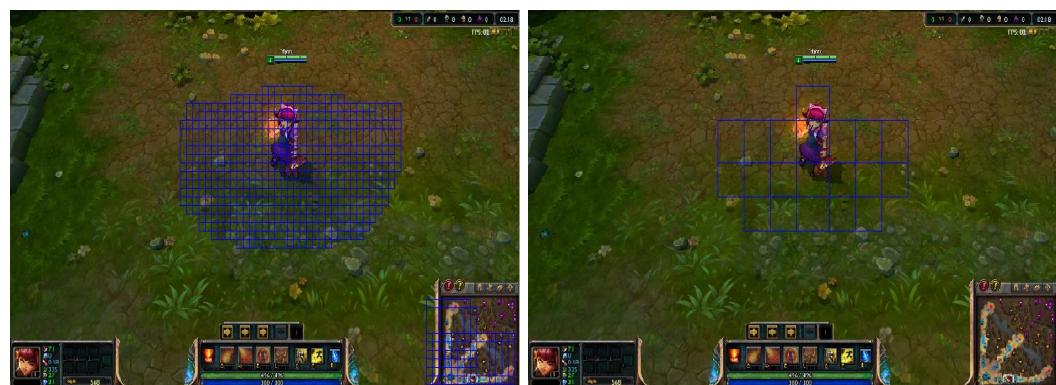


Fig. 4.3.5 y 4.3.6: Desp 10 y 40 con k = 5

3.4.- Conclusiones

Los resultados de esta aproximación empeoran incluso los obtenidos en la primera. Tras analizar la causa, se ha llegado a la conclusión a que es debido a que los patrones de color del pelo de Annie se repiten en más zonas del personaje, lo cual lleva a que se den positivos absurdos (detectar el peto como pelo, por ejemplo). La elección del rasgo no ha sido la adecuada. Se propone de alternativa considerar rasgos de mayor tamaño, o aumentar el número de los mismos.

Aproximación 4

4.1.-Contenido

La aproximación 4 presenta dieciocho características: las seis de la aproximación 1 y dos rasgos. El primer rasgo es la cabeza, y el segundo el cuerpo (esto es, partir la ventana en dos rectángulos desiguales).

4.2.- Recursos usados

A la hora de realizar la aproximación, han sido usadas las bases de datos ANNIE_STD_4_25.mat para los casos positivos y FONDO_STDHUD_4_25.mat para los casos negativos.

4.3.- Ejecuciones

A) Neuronas

El rasgo añadido parece surtir efecto. Aunque en desplazamientos pequeños sigue dando algunos problemas (falsos positivos con desp = 10), en general ofrece resultados similares a los de la aproximación 2, sobretodo en el caso de desp = 40.



Fig. 4.4.1 y 4.4.2: RNA para desplazamientos 40 y 10, respectivamente

B) SVM

Curiosamente, en SVM se perciben peores resultados en esta aproximación que en la anterior. Aunque RBF parece lograr resultados, MLP y Quadratic empeoran bastante. Un comportamiento decepcionante tras ver la mejoría en neuronas.

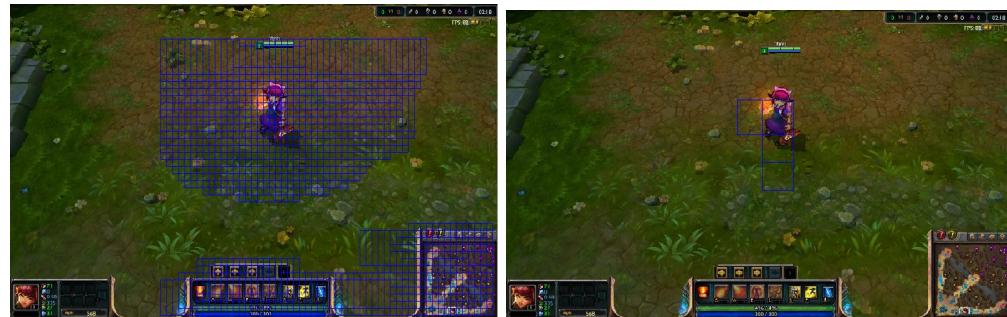


Fig. 4.4.3 y 4.4.4: MLP desp 10 y Polynomial desp 40

C) KNN

La aproximación 4 presenta resultados similares a la aproximación 2, esto es, peores que en el caso anterior. Vuelven a aparecer muchos falsos positivos en el HUD y ajusta peor a Annie. KNN no parece funcionar demasiado bien con estos rasgos.

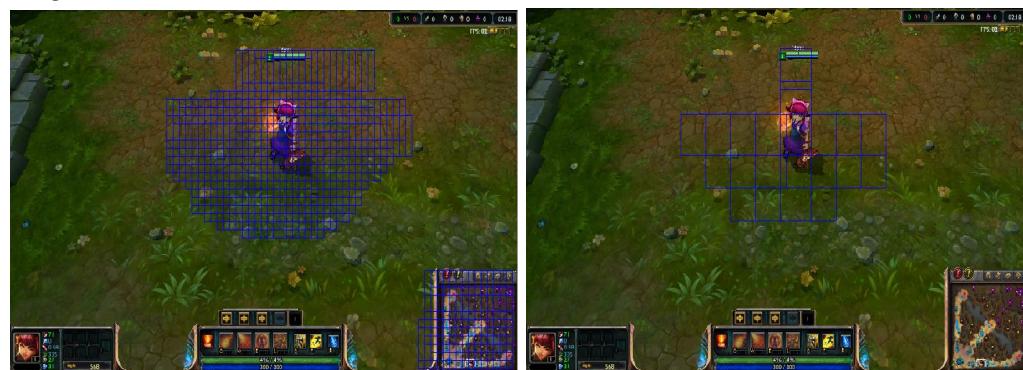


Fig. 4.4.5 y 4.4.6: Desp 10 para k = 5 y desp 40 para k = 1

4.4.- Conclusiones

Esta aproximación rivaliza con la dos en cuanto a mejores resultados obtenidos. La distinción cuerpo/cabeza es más significativa que el uso solo del rasgo pelo. Sin embargo, no cumple con las expectativas, ya que se esperaba una mejoría mucho mayor en general (KNN y SVM no mejoran con la 4). Es necesario encontrar otro rasgo (o rasgos) más identificativo.

Aproximación 5

5.1.-Contenido

La aproximación 5 presenta dieciocho características: las seis de la aproximación 1 y dos rasgos. El primer rasgo es el pelo, mientras que el segundo es la bola de fuego que Annie tiene en la mano (tercio superior una franja en la parte izquierda). Sin embargo, dicha bola de fuego no está presente en todas las animaciones, aunque si en las más comunes. Limitarse a aquellas en las que si esté la llama no es algo especialmente restrictivo, así que se aplica una nueva restricción a las capturas.

5.2.- Recursos usados

A la hora de realizar la aproximación, han sido usadas las bases de datos ANNIE_FRBSTD_1_25.mat para los casos positivos y FONDO_STDHUD_1_25.mat para los casos negativos.

5.3.- Ejecuciones

A) Neuronas

La aproximación cinco ofrece resultados prometedores. Es la primera aproximación con rasgos que no presenta falsos positivos. Logra reducir el número de ventanas con desplazamientos pequeños y marca un número muy reducido (en algunas ejecuciones solo una) en desplazamiento 40. En general, la mejor aproximación ejecutada hasta ahora con neuronas.



Fig. 4.5.1 y 4.5.2: RNA para desplazamientos 40 y 10, respectivamente

B) SVM

La mejora más drástica hasta la fecha. Todos los kernels ofrecen en algún desplazamiento resultados notables. Solo se han percibido falsos positivos en MLP, y los ajustes son mucho mejores que en cualquier otra aproximación. Mención especial a RBF, que ha aportado el mejor resultado obtenido en la práctica hasta el momento.



Fig. 4.5.3 y 4.5.4: RBF desp 10 y Quadratic desp 40

C) KNN

KNN no rompe la tónica de mejora. Aunque sigue estando por detrás de los otros clasificadores, se obtienen también en esta aproximación los mejores resultados hasta el momento para este sistema. El número de falsos positivos se reduce drásticamente, y se obtiene una precisión aceptable (aunque ni de lejos tan buena como en SVM y RNA).



Fig. 4.5.5 y 4.5.6: Desplazamiento 40 y 10 para k = 1

5.4.- Conclusiones

Sin duda, la mejor aproximación hasta la fecha. En algunas de las ejecuciones se consigue obtener de resultado una única ventana que encuadra de forma bastante aceptable al personaje.



Fig 4.c: mejor ejecución lograda en el trabajo (aproximación 5 con SVM con kernel RBF y desp 40)

Se considera que, al nivel al que se trabaja, se ha satisfecho los objetivos perseguidos en esta rama. En posteriores aproximaciones se intentará aumentar el número de fondos y personajes reconocidos.

Aunque hasta ahora se ha hablado de la calidad de los resultados, es importante no perder de vista la complejidad temporal:

- Entre clasificadores, neuronas es la técnica más lenta con diferencia (~30 veces peor que las otras técnicas). KNN y SVM están bastante igualados, pero gana SVM (~3 veces mejor)
- Como era previsible, la relación entre desplazamiento y tiempo es inversamente proporcional. El crecimiento del tiempo en función de lo reducido que sea el desplazamiento parece tener una tendencia exponencial, (se pasa de desplazamiento 10 en 30000 segundos a desplazamiento 25 en 160 segundos).
- Entre aproximaciones, a medida que se va añadiendo características y rasgos, son más costosas de procesar. La peor en tiempo es la cuatro, ya que requiere de procesar toda la ventana dos veces.

A la vista de los datos temporales obtenidos, y con todo lo que se ha comentado en las primeras cinco aproximaciones, se ha decidido prescindir de algunas ejecuciones. Se planteó el usar el kernel RBF con desplazamiento entre 30 y 40, ya que fue la técnica que mejores resultados aportó.

Para la siguiente aproximación se ha decidido que se cambiará a un nuevo dominio: Annie en diferentes fondos. Observando el arte conceptual del mapa clásico del juego, se puede ver que, además de las zonas verdes que se han utilizado, existen otras paletas de colores para suelos, como las zonas pedregosas de las esquinas (las “bases” de cada equipo), o los tonos azulados del río central, que son los que se van a considerar, manteniendo también los suelos de césped.



Fig. 4.c: Arte conceptual del mapa “Grieta del Invocador” del League of Legends

Realmente la aproximación seis consistió en probar todas las anteriores, modificando las bases de datos para incluir muestras de Annie en río, base y césped, así como los elementos característicos de cada localización. Debido a que en gran parte se repiten las observaciones realizadas entre las aproximaciones 1 a 5, se comentarán solo aquellos detalles nuevos y relevantes relacionados con el cambio de fondo.

Aproximación 6

6.1.-Contenido

La aproximación seis replica las características de las cinco anteriores:

- Medias RGB (3 características).
- Medias y desviaciones típicas RGB (6 características).
- Total y rasgo pelo. (12 características).
- Total y rasgos cabeza y cuerpo (18 características).
- Total y rasgo pelo y bola de fuego (18 características).

6.2.- Recursos usados

A la hora de realizar la aproximación, han sido usadas las bases de datos ANNIE_Y_X_25.mat para los casos positivos y FONDO_Y_X_25.mat para los casos negativos, donde Y es RIV o BAS y X es el una de las aproximaciones anteriores (de 1 a 5). En total, 20 bases de datos.

6.3.- Ejecuciones

A) Río

Los primeros resultados en río fueron bastante lamentables. La antorcha situada en la zona izquierda del mapeado causaba multitud de falsos positivos (sobre todo en KNN, donde la situación se disparaba). El problema fue un error al crear las bases de datos, ya que se mezclaron archivos por error. Una vez subsanado el problema, se obtuvieron resultados similares a los esperados, remarcando una vez más el buen funcionamiento de RBF con la aproximación 5.

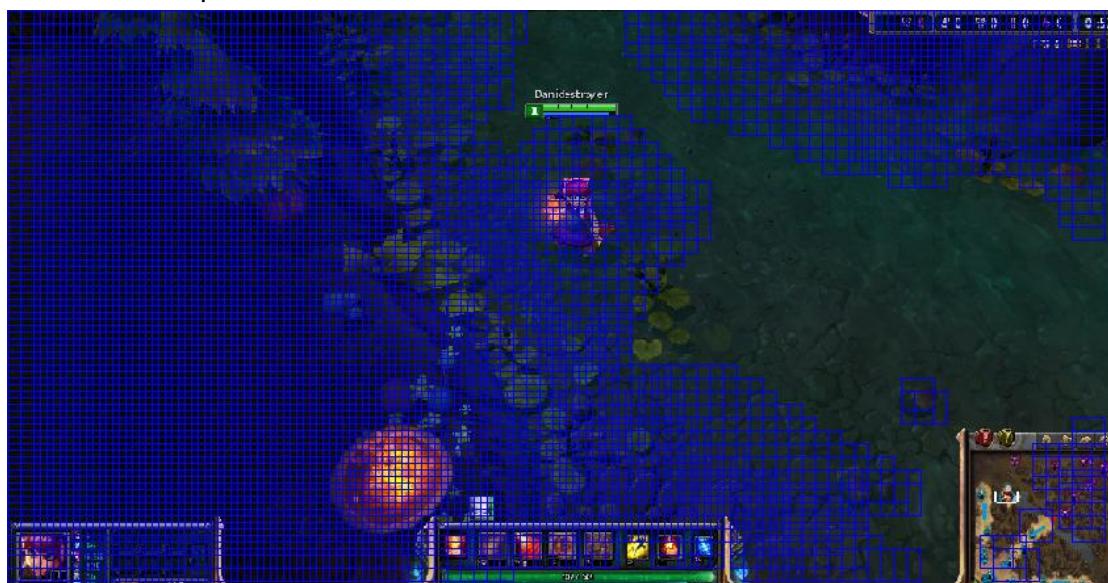


Fig 4.6.1: “resultado” con KNN y desplazamiento 15 con aproximación 1.



Fig 4.6.2: mejor resultado de río (aproximación 5 con kernel RBF y desplazamiento 25)

B) Base

A pesar de las diferentes estructuras de las bases, la detección en este fondo fue adecuada desde un primer momento. Los tonos grisáceos y azulados difieren bastante del esquema de colores de Annie, por lo cual, una vez más, se obtuvieron resultados similares a los de las anteriores aproximaciones.

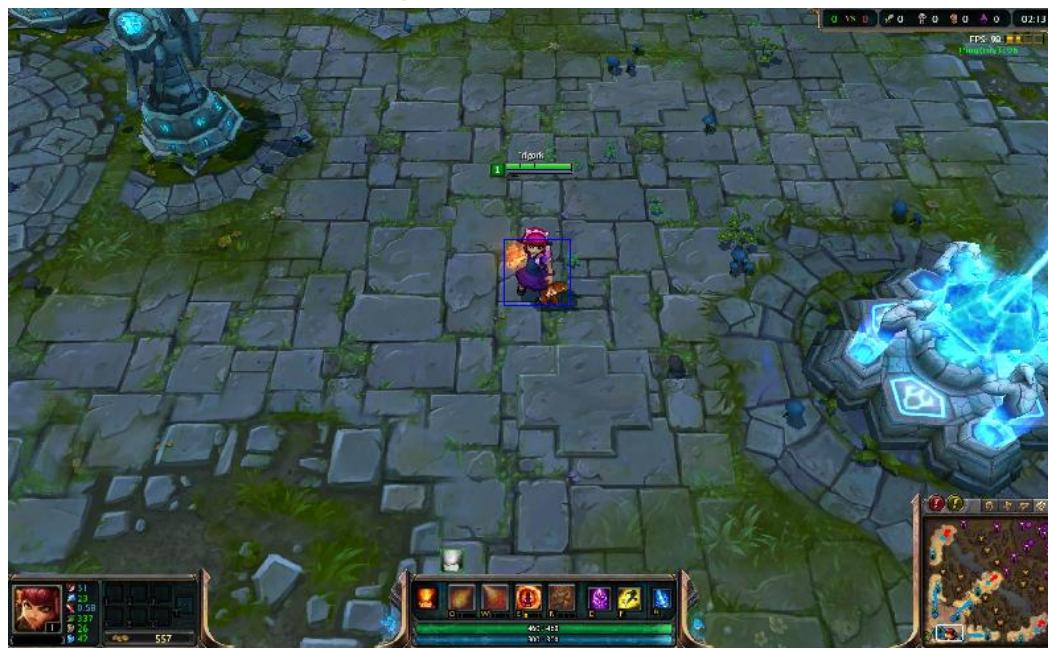


Fig 4.6.3: kernel RBF con desplazamiento 40 en base con la aproximación5

6.4.- Conclusiones

Se puede concluir que, quitando algún elemento conflictivo (la antorcha de río), el trabajo realizado parece funcionar bien con los distintos fondos del mapeado. Como ya se había comentado en la introducción, la diferencia de colores entre personajes y mapa es más que notable para lograr que los jugadores no “ pierdan” a sus personajes, confundiéndolos con la paleta de colores del escenario.

Llegados a este punto, se decidió ampliar el abanico de personajes, sin tener limitadas las restricciones de fondo. Se ha buscado trabajar con aquellos que tengan rasgos notables y diferenciadores. En concreto, las siguientes tres aproximaciones trabajarán con un nuevo personaje cada una, utilizando para ello únicamente máquinas de soporte vectorial con el kernel RBF, como ya se ha explicado.

Aproximación 7

7.1.-Contenido

Se utilizará al personaje llamado Zilean, con un rasgo, el pelo (12 características en total).



```
carac1 = image(round  
(sizey/3):sizey-round  
(sizey/2), round(sizex/2):2  
*round(sizex/3), 1:3);
```

Fig 4.7.1: Imagen de Zilean en la que se explica el rasgo usado

7.2.- Recursos usados

A la hora de realizar la aproximación, han sido usadas las bases de datos ZILEAN_ALL_7_75.mat para los casos positivos y FONDO_ALL_7_75.mat para los casos negativos.

7.3.- Ejecuciones



Fig 4.7.2, 4.7.3 y 4.7.4: Ejecuciones de Zilean en diferentes fondos, con desplazamiento 15

7.4.- Conclusiones

Debido a lo característico que resulta el pelo, el reconocimiento del personajes es realmente bueno desde el primer momento. Hay que tener en cuenta que, en sus animaciones por defecto, Zilean se mueve mucho, por lo cual partes llamativas, como las agujas del reloj, no pueden ser utilizadas como rasgo. De todos modos, la detección es lo suficientemente buena como para abordar otro personaje.

Aproximación 8

8.1.-Contenido

Se utilizará al personaje Urgot, con dos rasgos (la garra que forma su mano izquierda y una zona con suturas del abdomen, un total de 18 características).

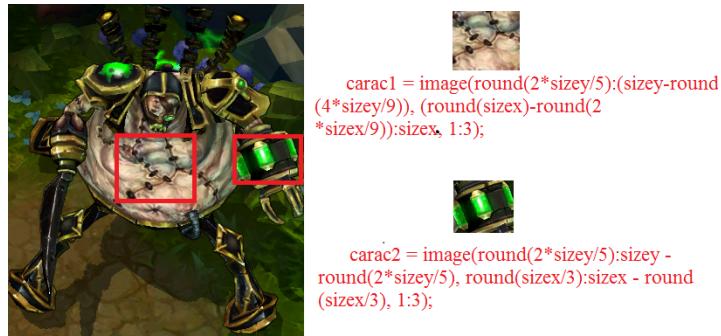


Fig 4.8.1: Urgot y sus rasgos

8.2.- Recursos usados

A la hora de realizar la aproximación, han sido usadas las bases de datos URGOT_ALL_8_75.mat para los casos positivos y FONDO_ALL_8_75.mat para los casos negativos.

8.3.- Ejecuciones



Fig 4.8.2, 4.8.3 y 4.8.4: Urgot en Base(desplazamiento 30), río (desplazamiento 15) y césped (desplazamiento 15).

8.4.- Conclusiones

De manera análoga al caso de Zilean, Urgot es reconocido de forma satisfactoria. El tono verde de su brazo es muy llamativo en relación a las paletas grisáceas y negras de su estómago. Al igual que en la aproximación anterior, rasgos como la cabeza o el otro brazo no son usados como referentes debido a que en muchas de sus animaciones aparecen en posiciones diferentes.

Aproximación 9

9.1.-Contenido

Se utilizará al personaje Thresh con dos rasgos (la zona con cadenas sobre la camisa azul y la linterna que porta en una mano, un total de 18 características).



Fig 4.9.1: Thresh y sus rasgos

9.2.- Recursos usados

A la hora de realizar la aproximación, han sido usadas las bases de datos THRESH_ALL_9_75.mat para los casos positivos y FONDO_STDHUD_9_75.mat para los casos negativos.

9.3.- Ejecuciones



Fig 4.9.2: Thresh en césped con desplazamiento 15

9.4.- Conclusiones

Thresh fue probado para tratar de ver si realmente el sistema presentaría problemas en casos en los que la media del personaje pudiera entrar en conflicto con la media de colores de alguno de los fondos. Los resultados quizá no son tan precisos como en los casos anteriores (la guadaña queda sin cubrir), pero esto es debido a que, al trabajar con capturas rectangulares, utilizar un encuadrado que no se ajuste lo máximo posible podría provocar falsos positivos. A pesar de que a nivel de pruebas fue testeado únicamente en el fondo de césped, en la aproximación final se trabajó con él en río y base, con buenos resultados.

Como parte final del desarrollo, se probó al sistema con una última aproximación. La idea era ver si sería capaz de reconocer a varios personajes en la misma imagen. Aunque puede parecer trivial en principio, al trabajar con SVM existe un problema, que es que la salida de estos clasificadores, al contrario que en las redes de neuronas, es binaria (esto es, decide entre dos clases posibles). El número de clases totales en la siguiente aproximación oscila entre 3 (personaje1, personaje2, fondo) y 5 (4 personajes y fondo). Para solucionar ese problema, se optó simplemente por utilizar las mejores aproximaciones de cada personaje a la vez, de tal manera que cada ventana es pasada por n SVM (siendo n el número de personajes), y recuadrando en diferentes colores la salida de cada uno. Además, a modo de guiño, se hizo un pequeño cambio al sistema para que localice la ventana más baja en el eje y de cada personaje y escriba su nombre debajo.

Aproximación 10

10.1.-Contenido

Se combinarán a Annie, Zilean, Urgot y Thresh, con las aproximaciones 5, 7, 8 y 9 (respectivamente) en diversas ejecuciones (Annie/Zilean, Annie/Zilean/Urgot y los cuatro en la misma imagen).

10.2.- Recursos usados

Se han utilizado las bases de datos indicadas en las aproximaciones 5, 7, 8 y 9, en función de los personajes que se hayan usado en cada imagen.

10.3.- Ejecuciones



Fig 4.10.1: Todos en río

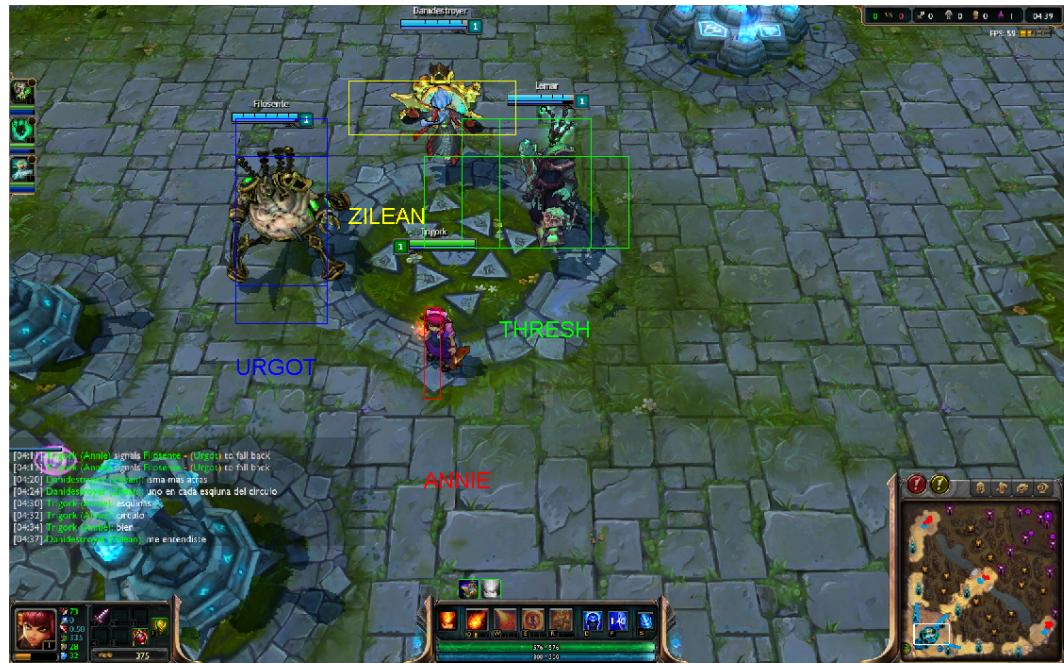


Fig 4.10.2: Todos en base



Fig 4.10.3: Todos en césped

10.4.- Conclusiones

Los resultados son realmente buenos. Ningún SVM produce falsos positivos, y los encuadrados son, quitando alguna anomalía, los que cada aproximación ofrecía ya anteriormente. La presencia de varios fondos no parece modificar sustancialmente el reconocimiento. En definitiva, se puede considerar la aproximación diez como el mejor resultado logrado en el desarrollo.

Para finalizar el apartado de desarrollo, se anexan las tablas de tiempos obtenidas al realizar todos los procesos descritos en las nueve primeras aproximaciones. La aproximación 10 fue realizada de manera ilustrativa y a modo de testeo y cierre, por lo cual no se aportan datos temporales sobre ella.

Base

RRNNA Despl \ Aprox		1	2	3	4	5
	10	11179,1573	11357,484	11454,9054	11737,1338	11595,1394
	25	368,2264	390,1985	393,7271	406,2159	402,3999
	40	70,5791	73,4653	71,628	72,7307	73,9944
<hr/>						
k	KNN Despl \ Aprox	1	2	3	4	5
1	10	1170,5849	1509,3418	1606,1249	1767,9683	1704,4944
	25	40,8938	49,4694	50,2643	56,5574	55,7206
	40	7,7354	9,0663	9,2088	10,4333	10,174
5	10	1199,8664	1523,3207	1659,7831	1779,5561	1719,8005
	25	41,7021	49,9025	50,3223	58,0438	55,8329
	40	7,7632	9,1924	9,3477	10,5209	10,2079
9	10	1204,1071	1530,9258	1594,2965	1786,2434	1724,5048
	25	41,3247	50,779	50,8877	57,3574	55,9649
	40	7,808	9,2207	9,2916	10,5517	10,0827
<hr/>						
krn	SVM Despl \ Aprox	1	2	3	4	5
lin	10	311,8412	503,3702	621,9543	822,7306	785,4742
	25	12,1738	19,6738	23,6742	30,523	28,8466
	40	2,3427	3,6944	4,4157	5,6589	5,4363
quad	10	313,6755	504,0145	624,6539	825,8952	776,07
	25	11,7858	19,7246	23,5881	30,9673	29,1719
	40	2,3169	3,7422	4,4246	5,7053	5,4365
poly	10	314,3375	506,5901	627,3872	833,0165	781,8097
	25	11,9844	20,0485	23,821	30,9441	29,357
	40	2,3667	3,7259	4,4998	5,7661	5,4846
rbf	10	379,0419	579,8726	690,8956	898,4627	846,5749
	25	14,4862	22,5032	26,3909	33,3556	31,693
	40	2,7938	4,1894	4,8796	6,1737	5,8546
mlp	10	323,3649	518,3016	625,1437	830,7134	778,499
	25	12,1671	20,3505	23,7707	30,7651	29,0744
	40	2,3914	3,7636	4,4661	5,8211	5,4836
<hr/>						
krn	SVM Despl \ Aprox	7	8	9		
rbf	10	179,3898	226,2321			
	25	11,4676	14,2463			
	40	3,0002	3,6925			

Tabla 1: tiempos obtenidos en las ejecuciones con las aproximaciones 1 a 9 para diferentes técnicas/desplazamientos (en segundos) en base

Río

RRNNAA Despl \ Aprox		1	2	3	4	5
	10	14485,462	14007,1188	13380,3173	13707,0972	13481,3847
	25	352,9031	359,1109	363,9885	370,5455	367,776
	40	72,6486	74,1352	74,7654	76,2235	75,8874
<hr/>						
k	KNN Despl \ Aprox	1	2	3	4	5
1	10	1386,5966	1653,469	1666,4703	1931,8719	1852,7597
	25	37,7834	43,9668	44,5898	51,2409	49,5627
	40	7,8489	9,2498	9,4818	10,807	10,4149
5	10	1412,6944	1671,6552	1689,3253	1950,3712	1868,1882
	25	38,1752	44,502	45,1176	52,4037	49,8393
	40	7,9917	9,3755	9,54	10,9318	10,4908
9	10	1413,6371	1741,3453	1685,5287	1950,9741	1869,2115
	25	38,2776	44,6622	45,1235	51,726	49,9855
	40	8,0115	9,42	9,604	10,9129	10,6007
<hr/>						
knn	SVM Despl \ Aprox	1	2	3	4	5
lin	10	399,7566	650,349	775,3215	1037,1178	962,1456
	25	10,9105	17,0376	20,6591	27,1097	25,6598
	40	2,6607	3,8805	4,5189	5,9979	5,8017
quad	10	492,3644	644,5062	775,2631	1044,542	962,277
	25	12,5213	16,9579	20,9101	27,398	25,3123
	40	2,7864	3,7387	4,5003	5,902	5,5496
poly	10	507,3611	648,1546	779,3402	1039,1921	968,2441
	25	12,7008	16,9682	20,6843	27,0788	25,499
	40	2,7727	3,7885	4,5604	5,8864	5,6002
rbf	10	470,9541	731,5353	852,4904	1118,4954	1044,9249
	25	12,7069	19,1497	22,5883	29,2352	27,6384
	40	2,8048	4,242	4,9198	6,4233	5,9979
mlp	10	396,8442	653,8395	777,241	1051,1076	967,1391
	25	10,7566	17,1458	20,5318	28,669	25,3918
	40	2,407	3,8578	4,5938	5,9855	5,5227
<hr/>						
knn	SVM Despl \ Aprox	7	8	9		
rbf	15	176,8278	222,9338			
	30	11,4494	14,2175			
	45	3,0769	3,7971			

Tabla 2: tiempos obtenidos en las ejecuciones con las aproximaciones 1 a 9 para diferentes técnicas/desplazamientos (en segundos) en río

Standard (Tierra/Césped)

RRNNAA Despl \ Aprox	1	2	3	4	5
10	5886,5143	6059,5671	6078,8686	6214,0791	6177,043
25	153,8768	157,4343	158,9077	162,4807	161,8576
40	31,0425	31,7605	32,1516	32,6311	32,5207

k	KNN Despl \ Aprox	1	2	3	4	5
1	10	600,5266	736,1761	749,985	881,2427	832,9917
	25	17,0482	20,6822	20,8965	24,4069	23,2571
	40	3,5778	4,2711	4,3224	5,0267	4,7954
5	10	616,3189	747,6376	754,836	888,5734	837,0605
	25	17,283	20,8334	21,0614	24,5354	23,4807
	40	3,5889	4,3385	4,4081	5,1448	4,8417
9	10	621,1459	753,3326	755,7807	891,6736	839,2639
	25	17,4238	20,9605	21,0868	24,7718	23,3718
	40	3,5702	4,3273	4,3567	5,1415	4,8735

krn	SVM Despl \ Aprox	1	2	3	4	5
lin	10	204,8973	324,7222	401,1654	537,6577	461,4001
	25	5,3273	8,9776	10,5477	14,0222	13,0263
	40	1,2743	1,9289	2,2587	3,0225	2,8291
quad	10	205,4035	335,7921	400,8306	538,2699	463,0102
	25	5,5181	8,8854	10,4217	14,0201	12,7468
	40	1,2976	1,9282	2,3191	3,0001	2,7335
poly	10	208,7122	342,5191	404,2422	538,9372	463,3932
	25	5,4642	8,9518	10,4867	14,0013	12,7561
	40	1,2723	1,9715	2,3525	3,0088	2,7542
rbf	10	242,8233	381,7737	443,0573	555,4772	497,6568
	25	6,3812	9,9678	11,5156	15,0277	13,7337
	40	1,4057	2,174	2,4955	3,2806	2,963
mlp	10	208,0988	353,6488	402,461	515,3275	464,3746
	25	5,3755	9,1236	10,492	14,446	12,7799
	40	1,255	2,0591	2,3153	3,4669	2,7549

krn	SVM Despl \ Aprox	6	7	9
rbf	15	143,9897	222,575	221,9944
	30	9,9997	14,4679	14,6475
	45	2,1858	3,6368	3,9006

Tabla 3: tiempos obtenidos en las ejecuciones con las aproximaciones 1 a 9 para diferentes técnicas/desplazamientos (en segundos) en césped.

5 Conclusiones

Tras cuatro meses de trabajo, el grupo de desarrollo considera que se han alcanzado unos resultados realmente buenos. Se ha logrado recuadrar, con una precisión más que aceptable, diversos personajes en muestras representativas de todo el mapeado, llegando a reconocer hasta cuatro personajes simultáneos en pantalla.

Con una diferencia abismal, el clasificador que mejores resultados ha aportado ha sido SVM con kernel RBF. Si bien RNA aportaba resultados también muy buenos, su naturaleza estocástica, unida a sus pésimos tiempos de ejecución, ha llevado al grupo de trabajo a descartarla. KNN ha aportado, en general, resultados bastante malos, tanto en tiempo como en precisión de reconocimiento.

La labor de desarrollo ha tenido como principales escollos la falta de conocimiento del equipo de trabajo de las tecnologías usadas al inicio del proyecto (tanto del propio Matlab como de la implementación de los clasificadores en el mismo) y el enfoque inicial. Una vez construído el sistema básico (capaz de generar las bases de datos, extraer las características, aplicar las aproximaciones y realizar el barrido por la imagen), el desarrollo ha sido fluído, centrado en el planteamiento de rasgos y la realización de pruebas con los mismos.

Llegados a este punto, cabe preguntarse si realmente sería aplicable este sistema en la problemática real planteada al inicio de este documento. La respuesta, actualmente, es que no. Los tiempos de ejecución impiden utilizar este proyecto para reconocimiento en tiempo real. Sin embargo, es un primer paso. El grupo de desarrollo considera que, de seguir investigando por las líneas que se plantean a continuación, si sería posible y viable cumplir con las ideas iniciales del trabajo.

6 Trabajo futuro

Si bien es cierto que se han obtenido unos resultados bastante considerables, hay ciertos aspectos que sería necesario trabajar para llegar a técnicas con utilidad real.

El gran problema, como ya se ha comentado, radica en el tiempo de ejecución. Es impensable, en un juego que funciona entre 30 y 60 imágenes por segundo, que el procesado de una única imagen tarde entre 3 segundos y 3 horas y media. La complejidad temporal radica, principalmente, en el uso de ventanas como método de búsqueda en la captura. Un método por búsqueda de una región concreta que puede ser candidata a contener un personaje, y que amplie la zona candidata a partir de ahí, reduciría muchísimo la cantidad de tiempo que se tarda en lograr un resultado. Otra opción podría ser la paralelización de código aprovechando las toolbox que Matlab ofrece para ello, ya que el procesamiento de ventanas no tiene ninguna restricción de orden.

Una continuación clara al trabajo pasaría por identificar rasgos de todos los personajes del juego (118 a la hora de escribir estas líneas). Aplicando las técnicas ya vistas, sería simplemente realizar un estudio de los esquemas de color, encontrar las regiones más diferenciables y aplicar los scripts utilizados hasta ahora.

En cuanto al propio reconocimiento, los canales RGB se quedan cortos en algunos casos. Utilizar nuevas aproximaciones (texturas, siluetas, bordes...) mejoraría bastante el resultado. Además, se podrían lograr técnicas independientes de rasgos, que mejorarían la capacidad del sistema de discriminar personajes, aunque estos estuvieran en posiciones diferentes.

Líneas de trabajo interesantes podrían ser tratar de reconocer los “ataques” de los personajes (que tienen una animación y partículas características), o tratar de identificar, en base a una escena, niveles de peligro, utilizando para ello elementos de la interfaz.

Realmente, el dominio presenta un amplio abanico de posibilidades, limitado únicamente por la inventiva, las técnicas conocidas y la capacidad tecnológica.

7 Bibliografía

¹ REDBEARD, Un campeonato mundial, 32 millones de espectadores (accesed May 12th 2014 at <http://euw.leagueoflegends.com/es/news/esports/esports-editorial/un-campeonato-mundial-32-millones-de-espectadores>)

² FISHER MATTHEW, Starcraft 2 Automated Player at (accesed May 12th 2014 at <http://graphics.stanford.edu/~mdfisher/GameAIs.html>)